REPORT DOCUMENTATION PAGE

1a Report Security Classification: Unclassified	1b Restrictive Markings
2a Security Classification Authority	3 Distribution Availability of Report: Approved for public release; distribution is unlimited.
2b Declassification Downgrading Schedule	
4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization: Naval Postgraduate School; 6b Office Symbol (*if applicable*): 570	7a Name of Monitoring Organization: Naval Postgraduate School
6c Address (*city, state, and ZIP code*): Monterey, CA 93943-5000	7b Address (*city, state, and ZIP code*): Monterey, CA 93943-5000
8a Name of Funding Sponsoring Organization; 8b Office Symbol (*if applicable*)	9 Procurement Instrument Identification Number
8c Address (*city, state, and ZIP code*)	10 Source of Funding Numbers: Program Element No; Project No; Task No; Work Unit Accession N

11 Title (*include security classification*): NUMERICAL FIELD MODEL SIMULATION OF FIRE AND HEAT TRANSFER I A RECTANGULAR COMPARTMENT

12 Personal Author(s): Kenneth J. Thorkildsen

13a Type of Report	13b Time Covered	14 Date of Report (*year, month, day*)	15 Page Count
Master's Thesis	From To	September 1992	195

16 Supplementary Notation: The views expressed in this thesis are those of the author and do not reflect the official policy or po sition of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms (*continue on reverse if necessary and identify by block number*)
Field	Group	Subgroup	Fire modeling, compartment fire simulation, computer modeling

19 Abstract (*continue on reverse if necessary and identify by block number*)

Shipboard fires have been the bane of mariners since man's earliest attempts to sail the sea. Understanding the behavi of fire in an enclosed space, such as those found on today's modern seagoing vessels, will greatly enhance the mariner's abili to combat or prevent them. In a joint effort between the Naval Postgraduate School and the University of Notre Dame computer code has been developed to model a full scale fire in a closed compartment. The code uses a finite volume fo mulation to obtain numerical solutions to the unsteadv, three dimensional conservation equations of mass, momentum ar energy. Included are the effects of turbulence, strong buoyancy, surface radiation and wall conduction. The code gives v locities, pressure, temperatures, and densities throughout the field.

This thesis applies that computer code to the U.S. Navy's full scale fire test chamber at Naval Air Warfare Center, Chir Lake, California. Advanced computer graphics techniques, including color contouring and three dimensional vector fie plotting have been applied to make output data more informative. It is hoped that someday this model could provide a usef tool for naval architects in the design of a fire safe ship, and a cost effective means for development/evaluation of ne firefighting equipment and techniques.

20 Distribution Availability of Abstract	21 Abstract Security Classification	
☒ unclassified/unlimited ☐ same as report ☐ DTIC users	Unclassified	
22a Name of Responsible Individual: M.D. Kelleher	22b Telephone (*include Area code*): (408) 646-2530	22c Office Symbol: ME/kk

DD FORM 1473,84 MAR — 83 APR edition may be used until exhausted; All other editions are obsolete

T259137

Approved for public release; distribution is unlimited.

Numerical Field Model Simulation
of
Fire and Heat Transfer
in a Rectangular Compartment

by

Kenneth J. Thorkildsen
Lieutenant, United States Coast Guard
B.S., United States Coast Guard Academy, 1980

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1992

ABSTRACT

Shipboard fires have been the bane of mariners since man's earliest attempts to sail the sea. Understanding the behavior of fire in an enclosed space, such as those found on today's modern seagoing vessels, will greatly enhance the mariner's ability to combat or prevent them. In a joint effort between the Naval Postgraduate School and the University of Notre Dame a computer code has been developed to model a full scale fire in a closed compartment. The code uses a finite volume formulation to obtain numerical solutions to the unsteatdy, three dimensional conservation equations of mass, momentum and energy. Included are the effects of turbulence, strong buoyancy, surface radiation and wall conduction. The code gives velocities, pressure, temperatures, and densities throughout the field.

This thesis applies that computer code to the U.S. Navy's full scale fire test chamber at Naval Air Warfare Center, China Lake, California. Advanced computer graphics techniques, including color contouring and three dimensional vector field plotting have been applied to make output data more informative. It is hoped that someday this model could provide a useful tool for naval architects in the design of a fire safe ship, and a cost effective means for development/evaluation of new firefighting equipment and techniques.

1360
c.1

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

I. INTRODUCTION 1
A. BACKGROUND 1
B. COMPUTER MODELING 2
C. THE FIRE TEST FACILITY 3
D. THE COMPUTER PROGRAM 8

II. THE NUMERICAL MODEL 10
A. THE GOVERNING EQUATIONS. 10
B. THE CONTROL VOLUME. 11
C. DISCRETIZATION OF THE CONSERVATION EQUATIONS. 13
1. The Continuity Equation. 13
2. The Energy Equation. 14
3. The Momentum Equations. 15
D. PRESSURE CORRECTIONS. 18
1. Global Pressure Corrections. 18
2. Local Pressure Corrections. 19
E. INITIAL AND BOUNDARY CONDITIONS. 21
1. Initial Conditions. 21
2. Boundary Conditions. 21
F. MODELING OF PHYSICAL PHENOMENA. 22
1. Wall Conduction. 22
2. Turbulence. 22
3. Radiation. 23

III. THE COMPUTER PROGRAMS 24
A. INTRODUCTION. 24
B. PROGRAM FIREBLD. 24
C. PROGRAM FIRE. 26
D. GRAPHICAL ANALYSIS. 27

IV. CONCLUSIONS AND RECOMMENDATIONS 49

A. CONCLUSIONS. 49
B. RECOMMENDATIONS. 49

APPENDIX A. PROGRAM FIREBLD 50

APPENDIX B. PROGRAM FIRE 61

APPENDIX C. PROGRAM ISOTHERM 150

APPENDIX D. PROGRAM VELOCITY 165

LIST OF REFERENCES 179

INITIAL DISTRIBUTION LIST 183

LIST OF TABLES

Table 1. INPUT DATA 25
Table 2. WALL DATA 26

LIST OF FIGURES

Figure 1. Fire test chamber at NAWC, China Lake, CA. 4
Figure 2. Plan view of fire test chamber at NAWC, China Lake, CA. 5
Figure 3. North south elevations of test chamber at NAWC, China Lake, CA. . . . 6
Figure 4. East west elevations of test chamber at NAWC, China Lake, CA. 7
Figure 5. Instrumentation of fire test chamber at NAWC, China Lake, CA. 9
Figure 6. One Dimensional Basic and Staggered Computational Cells. 12
Figure 7. Basic Cell Nomenclature on a Rectangular Grid. 13
Figure 8. Temperature Profile, Plan View, 30 seconds. 29
Figure 9. Temperature Profile, Plan View, 60 seconds. 30
Figure 10. Temperature Profile, Plan View, 90 seconds. 31
Figure 11. Temperature Profile, X-Z Profile, 30 seconds. 32
Figure 12. Temperature Profile, X-Z Profile, 60 seconds. 33
Figure 13. Temperature Profile, X-Z Profile, 90 seconds. 34
Figure 14. Temperature Profile, Y-Z Profile, 30 seconds. 35
Figure 15. Temperature Profile, Y-Z Profile, 60 seconds. 36
Figure 16. Temperature Profile, Y-Z Profile, 90 seconds. 37
Figure 17. Velocity Profile, Plan View, 30 seconds. 38
Figure 18. Velocity Profile, Plan View, 60 seconds. 39
Figure 19. Velocity Profile, Plan View, 90 seconds. 40
Figure 20. Velocity Profile, X-Z Profile, 30 seconds. 41
Figure 21. Velocity Profile, X-Z Profile, 60 seconds. 42
Figure 22. Velocity Profile, X-Z Profile, 90 seconds. 43
Figure 23. Velocity Profile, Y-Z Profile, 30 seconds. 44
Figure 24. Velocity Profile, Y-Z Profile, 60 seconds. 45
Figure 25. Velocity Profile, Y-Z Profile, 90 seconds. 46
Figure 26. Example of "Zoom" feature of ISOTHERM. 47
Figure 27. Example of "Zoom" feature of VELOCITY. 48

ACKNOWLEDGEMENTS

The author would like to take this opportunity to thank the following persons for their contributions to this work:

To Professor K.T. Yang of the University of Notre Dame, for providing the program FIRE which is central to this work and for making himself available whenever there were questions regarding the code.

To Mr. Kent Farmer of the Naval Air Warfare Center, China Lake, CA for providing the data regarding the fire test chamber.

To Dr. Devadatta Mukutmoni of the Naval Postgraduate School whose assistance with developement of the plotting programs and troubleshooting all of the programs was instrumental in completion of this work.

I. INTRODUCTION

A. BACKGROUND

Fire aboard ship is one of the professional mariner's worst fears. With no place to go to escape the heat and smoke the mariner must fight the blaze or face the loss of his vessel. Even though U.S. merchant marine and naval personnel receive some of the best fire fighting training available, annual losses to shipboard fires can easily run into the hundred million dollar range. Ship down time, equipment repair/replacement, personnel injuries and casualties all contribute to these costs and result in the degradation of our merchant marine and naval forces. To minimize these losses it is imperative that the phenomena of fire be studied and understood especially in the relatively closed environment found aboard ship. It is only through study and understanding that adequate means may be developed to prevent or mitigate the devastating effects of a shipboard fire.

Fire is a complex phenomena whose study is equally complex, requiring the combined knowledge of a variety of fields including fluid dynamics, heat and mass transfer, and combustion. Research into the mechanics of fire and development of methods to predict it's behavior will aid engineers in reducing the probability of its ignition and minimizing its effects once ignited. [Ref. 1]

It is the complexity of fire which makes its study so difficult, especially in a shipboard environment. Small scale fire studies have proven inadequate for predicting the behavior of large scale fires. Therefore, full scale studies are generally the only means for conducting realistic experimentation. However, such full scale experimentation can be very costly and dangerous. Shipboard fires often occur in fully enclosed, airtight spaces where pressures can build up during the fire. These spaces may have restricted accesses, contain electronic equipment, flammables and/or toxics, all of which add to the expense and danger of full scale experimentation. Additional costs will also be incurred as air quality and emission standards are stiffened across the country.

To study the phenomena of shipboard fire, the U.S. Navy has built a test facility at the Naval Research Laboratory in Washington, DC known as Fire-1. This facility is essentially a large cylindrical pressure vessel with spherical endcaps intended for use in full scale modeling of fires inside submarines and/or in closed airtight compartments aboard ships. Missile attacks against British warships during the Falklands crisis and the

Iraqi missile attack on the USS STARK in the Persian Gulf in 1987 prompted the U.S. Navy to build a second fire test facility at the Naval Air Warfare Center (NAWC) in China Lake, CA. The purpose of this facility is to study the effects of fire in a vented compartment [Ref. 2, pp. 7-8]. It is this facility which is modeled in this thesis.

A less expensive, and less dangerous alternative to full scale fire experimentation is the use of computer modeling techniques. The development of high speed, high capacity computers has allowed researchers to model thoroughly the complex fire phenomena and predict fire behavior without the expense of full scale testing. A properly developed computer model, validated against actual full scale test data, can provide a less expensive and safe alternative to full scale experimentation. Furthermore, the inherent flexibility of computer modeling allows it to be used on increasingly more complex geometries. Someday this may lead to modeling of entire ships during the design phase to identify areas particularly susceptible to fire, or for the accurate prediction of the effectiveness of new firefighting techniques.

B. COMPUTER MODELING

Field modeling uses finite difference techniques to subdivide the volume being studied, in this case the simulated shipboard compartment, into small, finite volume elements. Using initial conditions specified by the user and the finite difference forms of the equations for conservation of mass, momentum, and energy, values of temperature, pressure, velocity and density are calculated for each of the individual volumetric elements at discrete time intervals. Additional modeling of physical effects such as radiation, turbulence and wall conduction are included to increase the validity of the simulation. The enormous number of calculations necessary for this type of modeling mandates the use of large amounts of computer memory and high speed processors.

The basis for this thesis is provided by a large number of previous research projects. One of the earliest successful models of this type was developed by Aziz and Hellums [Ref. 3] at Rice University in Houston, Texas. They expressed the Navier-Stokes equations in terms of vorticity and vector potential, then solved the resulting three dimensional finite difference equations using a combination of the alternating direction method and successive over-relaxation (SOR). Later work by Mallinson and de Vahl Davis [Ref. 4], Morrison and Tran [Ref. 5]; Chan and Banerjee [Ref. 6]; Ozeo, Fujii, Lior and Churchill [Ref. 7]; and a host of others have all expanded on the use of finite difference techniques to model convective heat transfer in closed compartments.

In the late 1970's and early 1980's, R.G. Rehm and H.R. Baum began developing equations to describe the buoyant flow induced by large scale fires [Refs. 8, 9, 10 and 11] Work done at the University of Notre Dame used a two dimensional finite difference field model to predict velocities, temperatures and smoke concentrations in aircraft cabin fires [Ref. 12 and 13]. The development of a two dimensional model of transient, natural convection cooling was developed, and experimentally verified, by Nicolette *et al* [Ref. 14] using a semi-implicit upwind differencing scheme and global pressure corrections. Still more studies [Refs. 15, 16, 17, 18 , & 19] have utilized finite difference methods to solve non-linear, three dimensional partial differential equations for rectangular enclosures.

C. THE FIRE TEST FACILITY

A full scale test facility has been constructed at the Naval Air Warfare Center (NAWC) at China Lake, CA. This facility, designed to simulate full scale shipboard compartments, consists of three chambers [see Figure 1 on page 4 and Figure 2 on page 5]. The main chamber measures 20 feet by 20 feet by 10 feet and is vented to the atmosphere through an opening in a side wall, in a manner intended to simulate the penetration of a missile or other projectile. The second chamber measures 15 feet by 15 feet by 10 feet is located adjacent to the main chamber to the east, while the third chamber, also measuring 15 feet by 15 feet by 10 feet, is located on top of the main chamber. These secondary chambers are intended to provide a means to study the vertical and horizontal heat transmission rates. [Ref. 2, pp. 7-10]

The entire test structure is constructed of 3/8 inch thick steel bulkheads/walls reinforced by 5 inch I-beams on 5 foot centers, and 1/2 inch thick steel decks reinforced underneath by 12 inch I-beams on 5 foot centers. Additional support was provided for the overhead in the main chamber by several 5 inch I-beam columns. These columns were intended to eliminate any sagging and/or distortion of the overhead as it is repeatedly cycled from ambient conditions to the extreme temperatures expected during the missile fuel tests. Access to all three compartments is through hatches which open to the outside of the structure and are kept closed during all testing. There are no openings between any two adjacent chambers. The ventilation opening is located on the north face of the main chamber to simulate an impact hole, such as a missile strike [see Figure 3 on page 6 and Figure 4 on page 7]. [Ref. 2, pp. 10-12]

Instrumentation is provided within the test chamber to measure radiation heat flux, total heat flux, compartment bulk pressure, wall temperatures and compartment gas

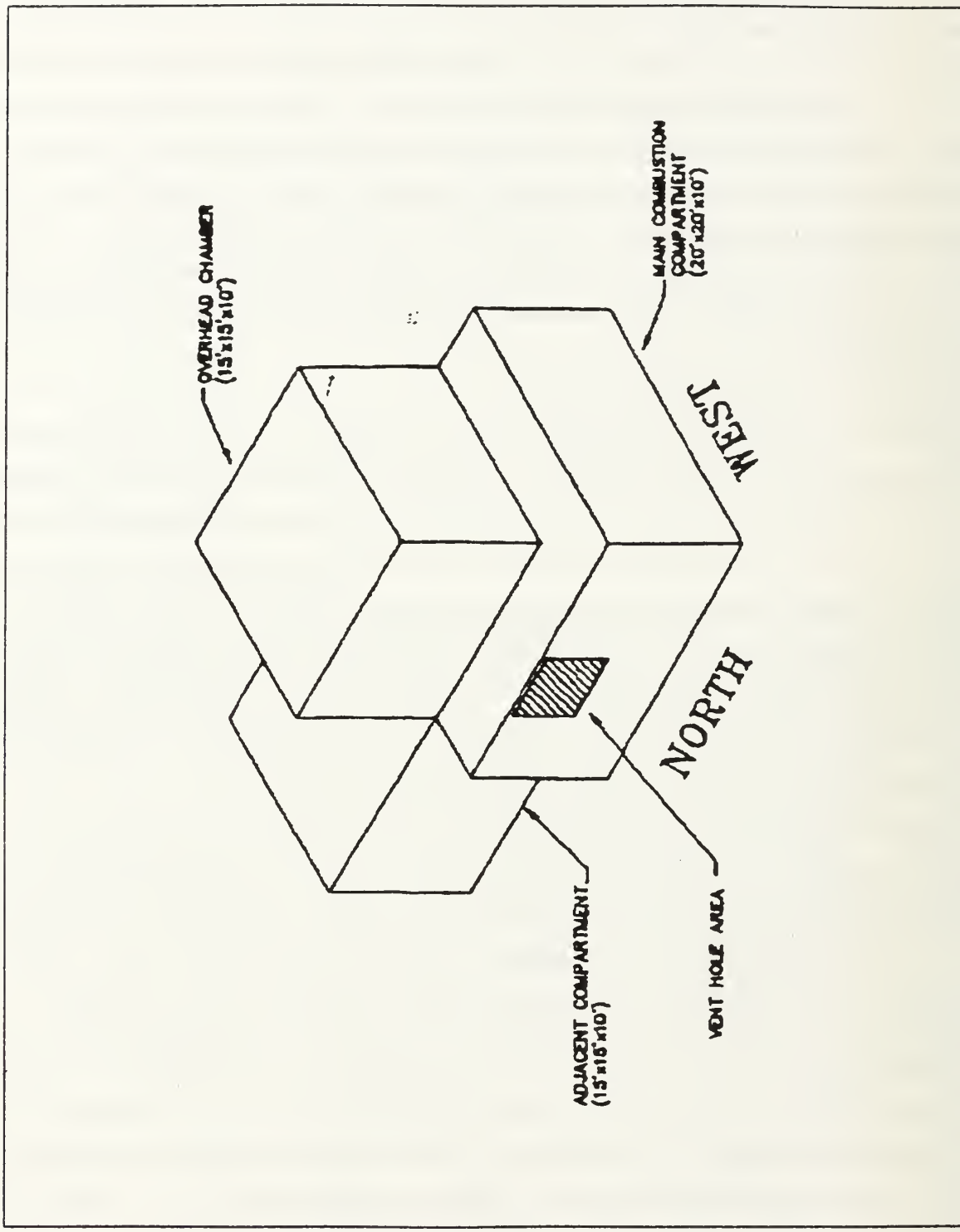

Figure 1. Fire test chamber at NAWC, China Lake, CA.

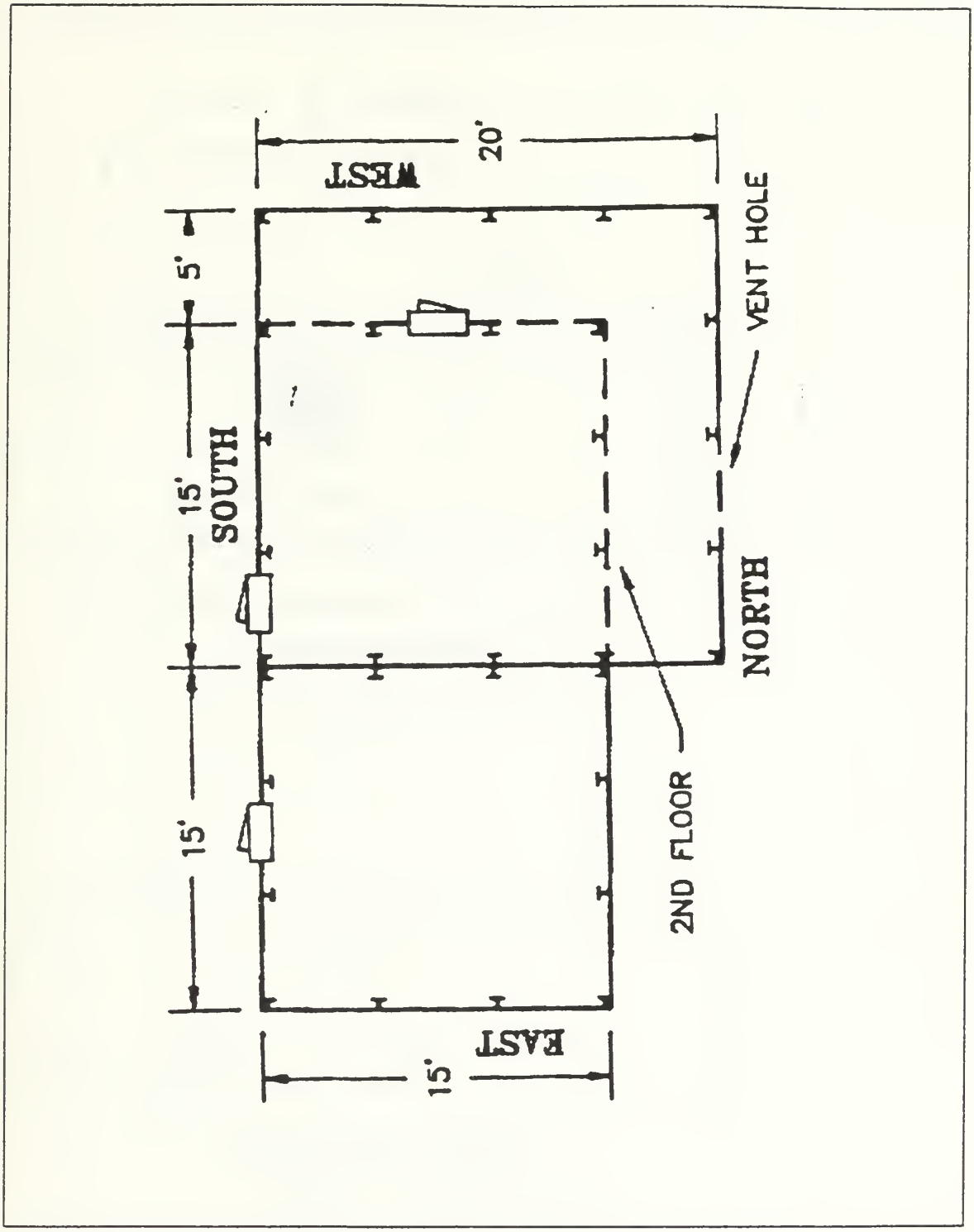

Figure 2. Plan view of fire test chamber at NAWC, China Lake, CA.

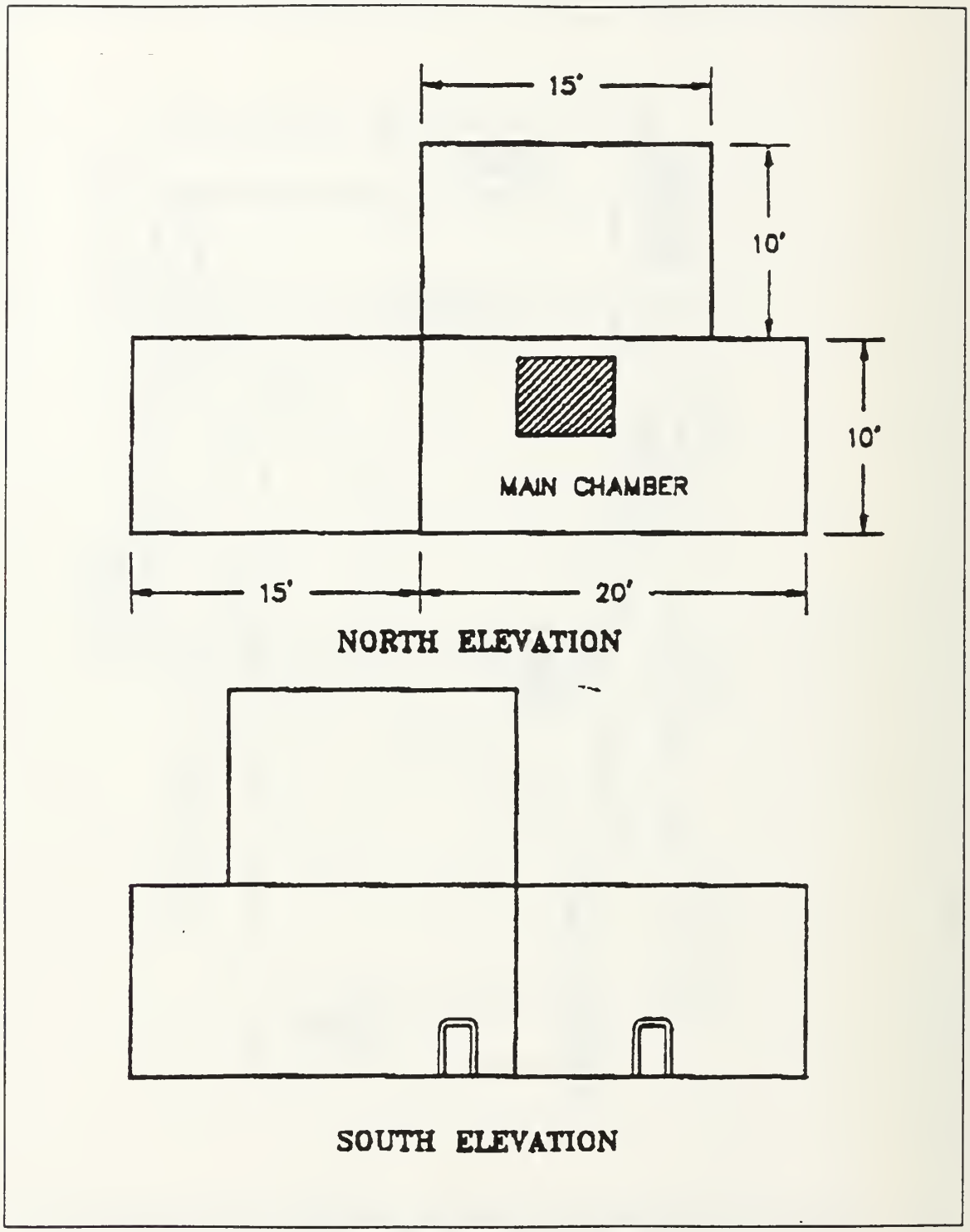

Figure 3. North/south elevations of test chamber at NAWC, China Lake, CA.

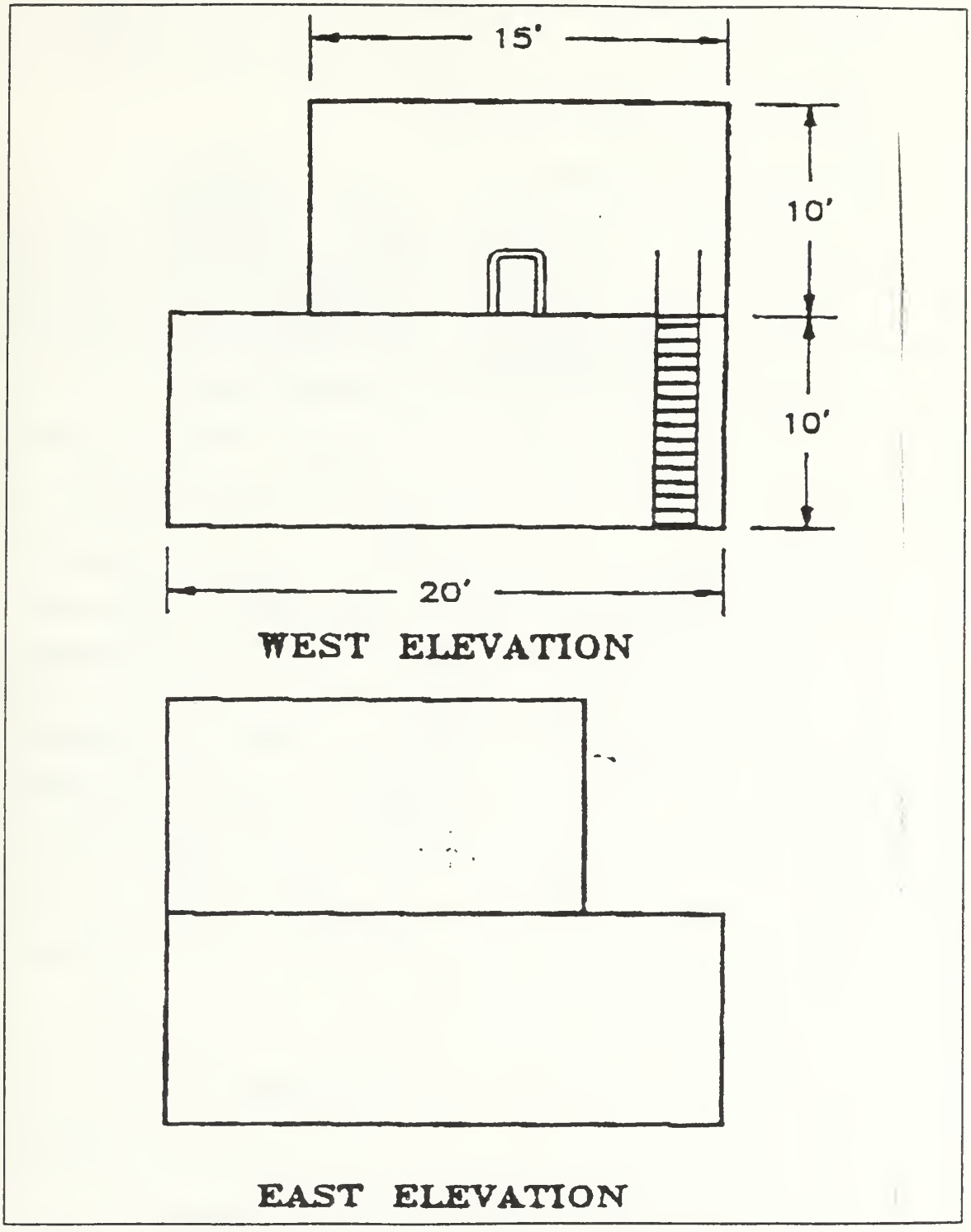

Figure 4. East/west elevations of test chamber at NAWC, China Lake, CA.

temperatures. Thermocouples were arranged in vertical arrays of 10 thermocouples. The lowest thermocouple in each array was placed 6 inches above the deck with subsequent thermocouples spaced at 1 foot intervals ending 6 inches below the ceiling [see Figure 5 on page 9]. [Ref. 2, pp. 14-15]

D. THE COMPUTER PROGRAM

This computer model is a joint project undertaken by the Naval Postgraduate School and the University of Notre Dame. It represents a low cost, safe alternative to full scale fire testing. With proper modifications, and properly validated by full scale experimentation, this program should provide a valuable tool for testing the effectiveness of fire mitigation techniques and evaluation of new ship designs.

Work on this program began at Naval Postgraduate School in 1986 when Nies [Ref. 20] used a cartesian coordinate system to model the cylindrical spherical geometry of the FIRE-1 test facility. In 1987 Raycraft [Ref. 21] modified the program to use a cylindrical spherical coordinate system. She also expanded the scope of this project by writing a companion program to calculate view factors and account for surface radiation effects. Using both of Raycraft's programs, Houck [Ref. 22] further expanded the scope of the project to account for the internal ventilation capabilities of the FIRE-1 facility. Most recently, in 1991, McCarthy [Ref. 23] used advanced computer graphics techniques to provide a more accurate pictorial representation of the unique geometry of the FIRE-1 facility and demonstrated the advantages of using full color displays to represent the three dimensional isotherm and velocity vector field profiles. This development greatly enhanced the presentation of the program output. McCarthy was also the first researcher at NPS to incorporate Raycraft's radiation/view factor program as subprograms of the main computer code.

This thesis returns to the cartesian coordinate system used by Nies [Ref. 20], to model the Navy's newest fire test facility at the Naval Air Warfare Center (NAWC) at China Lake, CA. It also advances McCarthy's [Ref. 23] work by applying the enhanced graphics capabilities of the NCAR Graphics software, developed by the National Center for Atmospheric Research [Refs. 24 and 25], for output presentation.

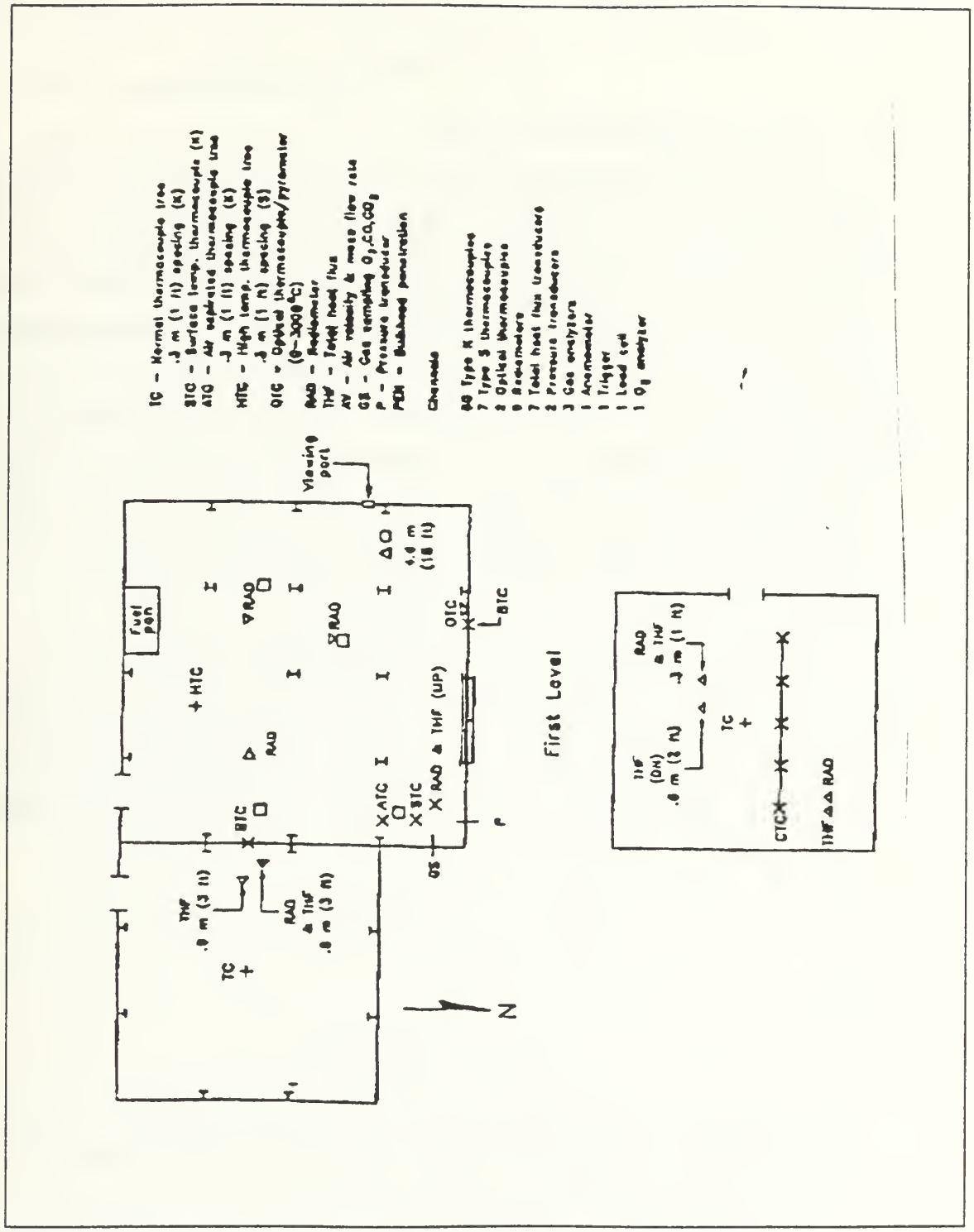

Figure 5. Instrumentation of fire test chamber at NAWC, China Lake, CA.

II. THE NUMERICAL MODEL

A. THE GOVERNING EQUATIONS.

The model used in this code is built on the fundamental laws of conservation of mass (continuity), energy and momentum. One example of these equations is presented by Patankar [Ref. 26, pp. 11-17]. As in that work, this model neglects the pressure work and viscous dissipation terms of the energy equation due to the low flow velocities expected. The volumetric heat generation term will be used to account for heat input by the fire. Patankar's version of the momentum equations [Ref. 26, p. 14, eq. 2.11] included both a body force and an extra viscous force. This model assumes a newtonian fluid therefore the extra viscous force is neglected and the only body force present is due to gravitational effects acting in the negative z-direction. Thus the body force in the z-momentum equation is set equal to $-\rho g$ where g is the local gravitational constant; the negative sign indicates that g is acting in the negative z-direction; and ρ is the fluid density.

Nies [Ref. 20, pp. 16-38] expanded the equations presented by Patankar into the three dimensional, cartesian coordinate system used in this model. Following the development pattern of Doria [Ref. 18, pp. 4-7] and making the assumption that air is a homogeneous gas of constant composition (reactive variations due to the fire are neglected), Nies [Ref. 20, pp. 16-17] incorporated the equations of state for an ideal gas with constant specific heats in the form:

$$P = \rho RT$$

and

$$h = c_p(T - T_{ref})$$

where P, T and ρ represent the bulk pressure, temperature and density inside the control volume, R is the gas constant for air, h is the specific enthalpy, c_p is the specific heat and T_{ref} is a suitably chosen reference temperature.

Again, following the procedures developed by Patankar [Ref. 26] and expanded by Doria [Ref. 18], Nies [Ref. 20 , pp. 21-26] goes on to place these six equations in their non-dimensional, integral forms. After subdividing the test chamber into a large, but

finite number of control volumes, finite difference techniques are used to solve for the six unknowns of temperature, pressure, density and the three components of velocity.

B. THE CONTROL VOLUME.

Each control volume, or cell, is defined by the nodal point contained at its center. Values for temperature, density and pressure are calculated at this central point (designated P) and are then assumed to hold for the entire cell. A secondary, or staggered grid system, is used to determine velocities. This staggered grid is offset from the main grid by one half the length of the cell [see Figure 6 on page 12]. As explained in McCarthy [Ref. 23, pp. 18-19] and Patankar [Ref. 26, pp. 115-120], use of this staggered grid alleviates two problems: first, since velocities are dependent upon pressure differentials, the staggered grid allows pressures to be determined at a more frequent interval thereby reducing the error associated with larger separations between nodal points; second, the stagger increases stability by decreasing and/or eliminating unrealistic, oscillatory velocity fields when adjacent velocities are used to satisfy the continuity equation.

Since Patankar's method uses primitive variables, in lieu of the stream function and vorticity, special attention must be paid to the coupling of equations through the pressure terms. An iterative process is used to calculate the pressure in each cell, then a local pressure correction is calculated to ensure that continuity is satisfied. Additional discussions of this correction are included in both Patankar [Ref. 26, pp.120-128] and Doria [Ref. 18, pp. 26-32]. A global pressure correction was described by Nicolette, *et al* [Ref. 14] for addressing net energy changes within the closed compartment. This correction has also been incorporated into this model.

As stated in McCarthy [Ref. 23, p. 19], forcing convergence on a non-linear set of equations, like those describing fluid flow, can be difficult. A variety of schemes have been developed, each with its own set of strengths and weaknesses. This model applies iterative techniques to a solution scheme known as "QUICK", or Quadratic Upstream Interpolation for Convective Kinematics, developed by Leonard [Ref. 19]. With the accuracy of a central differencing scheme and the stability of the convective diffusion terms in upwind differencing, the QUICK scheme estimates values and gradients of the transport variables at the cell faces. The utility of this method was demonstrated by H.Q. Yang [Ref. 27] when he used QUICK to solve the coupled momentum and energy equations for three dimensional flow in a tilted rectangular enclosure.

In addition to the center cell described above, neighboring cells will be used in various computations. To keep track of these cells a standardized nomenclature must be

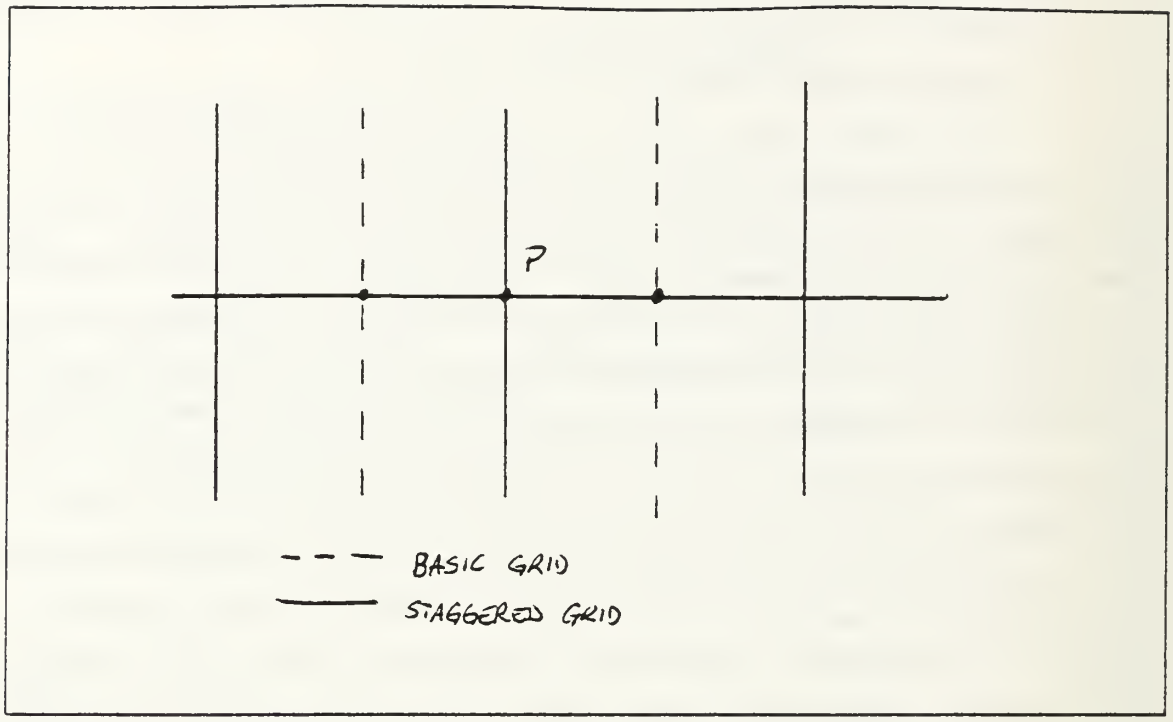

Figure 6. One Dimensional Basic and Staggered Computational Cells.

adopted. Assuming the central cell, cell P, to be located at the point (i,j,k) where i,j and k correspond to the standard coordinated axis x, y and z respectively, then each neighbor may be described as follows (NOTE: the following directions are for standardization of nomenclature only and do not necessarily correspond to compass directions as shown in Figure 1 on page 4):

East (i + 1,j,k)

West (i-1,j,k)

North (i,j + 1,k)

South (i,j-1,k)

Front (i,j,k + 1)

Back (i,j,k-1)

The nodal point in each direction is designated by the capital letter corresponding to the direction (i.e. E, W, N, S, F, B), while the boundary between cell P and its neighbors is designated by a lower case letter corresponding to that direction (i.e. the boundary between cell P and cell N is designated n, and the boundary between P and F is f). A typical cell array on a rectangular grid is shown in figure Figure 7 on page 13.

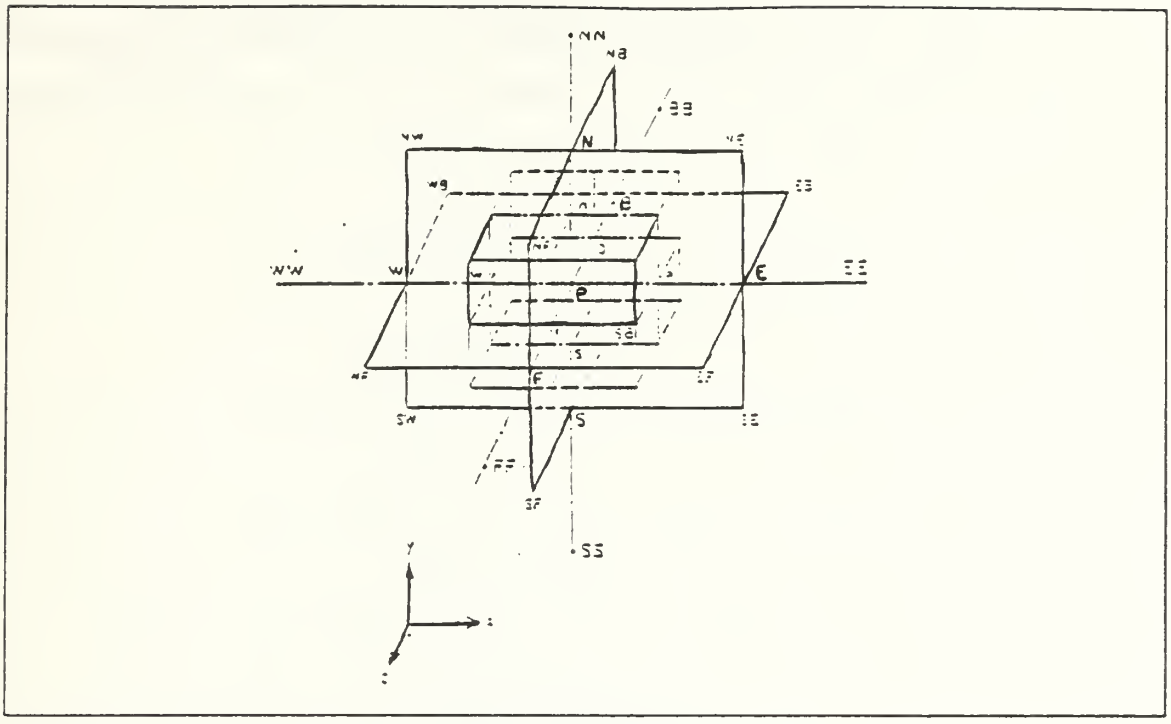

Figure 7. Basic Cell Nomenclature on a Rectangular Grid.

As discussed previously, each P node is used to determine values of density, pressure and temperature which are them applied to its entire cell. Velocities are determined at the cell faces using the staggered grid arrangement described above.

C. DISCRETIZATION OF THE CONSERVATION EQUATIONS.

With that brief background, the integral forms of the conservation equations developed Nies [Ref. 20, pp. 25-26] may now be discretized. As discussed by Nies [Ref. 20, pp.26-38] maximum stability and accuracy of the model is achieved by using three different finite differencing schemes. Forward differencing is used for timewise discretization, central differencing is used to discretize the diffusion terms, and the QUICK scheme is used to discretize the convective terms. Use of these techniques to discretize the governing equations was discussed in detail by Nies [Ref. 20, pp 26- 38] and will not be repeated here. The following finite difference forms of the governing equations are taken from his work.

1. The Continuity Equation.

$$(\rho_P - \rho_P^0)\frac{\Delta x \Delta y \Delta z}{\Delta t} + (G_e - G_w)\Delta y \Delta z + (G_n - G_s)\Delta z \Delta x + (G_f - G_b)\Delta x \Delta y = S_{mp}$$

where ρ_P is the density at the current time step; ρ_P^0 is the density at node P at the previous time step; Δt is the incremental time step; Δx, Δy, and Δz are the cell dimensions in the indicated direction; and S_m^p is the residual mass term. The mass flux (G) terms are defined as:

$$G_e = \frac{(\rho_E + \rho_P)}{2} u_e$$

$$G_w = \frac{(\rho_P + \rho_W)}{2} u_w$$

$$G_n = \frac{(\rho_N + \rho_P)}{2} v_n$$

$$G_s = \frac{(\rho_P + \rho_S)}{2} v_s$$

$$G_f = \frac{(\rho_F + \rho_P)}{2} w_f$$

$$G_b = \frac{(\rho_P + \rho_B)}{2} w_b$$

It should be noted that in all of these equations ρ refers to the density; u, v, and w are the three velocity components in the x, y, and z-directions respectively; uppercase subscripts denote values at the indicated nodal point; and lowercase subscripts denote values at cell faces. Also, in order for continuity to be satisfied in this closed system, the residual mass term (S_{mp}) would equal zero. However, due to the approximation inherent in the numerical scheme, we will be satisfied if S_{mp} tends toward zero as determined by comparison to an arbitrarily small threshold value.

2. The Energy Equation.

$$\left[H_{A_P} + \rho_P^0 \frac{\Delta x \Delta y \Delta z}{\Delta t} \right] h_P = H_{A_E} h_E + H_{A_W} h_W + H_{A_N} h_N + H_{A_S} h_S + H_{A_F} h_F + H_{A_B} h_B + H_{S_P}$$

where h is the specific enthalpy at the current time step; h^0 is the specific enthalpy at the previous time step; and the coefficients (H) are defined as:

$$H_{A_E} = \frac{(|G_e| - G_e)}{2} \Delta y \Delta z + \left(\frac{1}{Re_t Pr_t} \right)_e \frac{\Delta y \Delta z}{\Delta x}$$

$$H_{A_W} = \frac{(|G_w| + G_w)}{2}\Delta y\Delta z + \left(\frac{1}{Re_t Pr_t}\right)_w \frac{\Delta y\Delta z}{\Delta x}$$

$$H_{A_N} = \frac{(|G_n| - G_n)}{2}\Delta z\Delta x + \left(\frac{1}{Re_t Pr_t}\right)_n \frac{\Delta z\Delta x}{\Delta y}$$

$$H_{A_S} = \frac{(|G_s| + G_s)}{2}\Delta z\Delta x + \left(\frac{1}{Re_t Pr_t}\right)_s \frac{\Delta z\Delta x}{\Delta y}$$

$$H_{A_F} = \frac{(|G_f| - G_f)}{2}\Delta x\Delta y + \left(\frac{1}{Re_t Pr_t}\right)_f \frac{\Delta x\Delta y}{\Delta z}$$

$$H_{A_B} = \frac{(|G_b| + G_b)}{2}\Delta x\Delta y + \left(\frac{1}{Re_t Pr_t}\right)_b \frac{\Delta x\Delta y}{\Delta z}$$

$$H_{A_P} = H_{A_E} + H_{A_W} + H_{A_N} + H_{A_S} + H_{A_F} + H_{A_B}$$

$$H_{S_P} = \rho_P^0 h_P^0 \Delta x\Delta y\Delta z$$

Also used above are the turbulent Reynolds number (Re_t) and the turbulent Prandtl number (Pr_t). These are defined as

$$Re_t = \frac{\rho_0 u_0 H}{\mu_{eff}}$$

$$Pr_t = \frac{\mu_{eff} c_{p0}}{k_{eff}}$$

where H is the characteristic length (defined as the height of the test chamber in our model); ρ and c_p represent the density and specific heat of air, while the subscript 0 indicates that the properties are to be evaluated at the initial conditions which existed prior to ignition of the fire; u_0 is the reference velocity (set equal to 1.0 ft/sec); and μ_{eff} and k_{eff} are effective values of viscosity and conductivity as defined by Nies [Ref. 20, p. 39-40].

3. The Momentum Equations.

As stated by Nies [Ref. 20, p. 31] the momentum equations are more complex than the previous equations because of the use of the staggered grid and the addition of the shear stress terms. Staggered grids are determined by shifting the main grid one half cell in the negative direction along one axis at a time. Maintaining the nomencla-

ture established for the basic, centered cells, the central node of a cell staggered in the x-direction becomes w, the central node in a cell staggered in the y-direction becomes s, and the central node in a cell staggered in the z-direction becomes b. Likewise, the faces of the staggered cell are designated by the capital letters representing the basic nodes through which they pass. Thus, in the staggered grid P always represents the positive face along the axis where the shift (stagger) is being evaluated. Thus, the x-momentum equation becomes:

$$\left[A_w + \rho_w^0 \frac{\Delta x \Delta y \Delta z}{\Delta t}\right] u_w = A_e u_e + A_{ww} u_{ww} + A_{Nw} u_{Nw} + A_{Sw} u_{Sw} + A_{Fw} u_{Fw} + A_{Bw} u_{Bw} + S_w$$

where

$$A_e = \left[\frac{(|G_P| - G_P)}{2} + \frac{(\frac{1}{Re_t})_P}{\Delta x}\right] \Delta y \Delta z$$

$$A_{ww} = \left[\frac{(|G_{ww}| + G_{ww})}{2} + \frac{(\frac{1}{Re_t})_{ww}}{\Delta x}\right] \Delta y \Delta z$$

$$A_{Nw} = \left[\frac{(|G_{nw}| - G_{nw})}{2} + \frac{(\frac{1}{Re_t})_{nw}}{\Delta y}\right] \Delta z \Delta x$$

$$A_{Sw} = \left[\frac{(|G_{sw}| + G_{sw})}{2} + \frac{(\frac{1}{Re_t})_{sw}}{\Delta y}\right] \Delta z \Delta x$$

$$A_{Fw} = \left[\frac{(|G_{fw}| - G_{fw})}{2} + \frac{(\frac{1}{Re_t})_{fw}}{\Delta z}\right] \Delta x \Delta y$$

$$A_{Bw} = \left[\frac{(|G_{bw}| + G_{bw})}{2} + \frac{(\frac{1}{Re_t})_{bw}}{\Delta z}\right] \Delta x \Delta y$$

$$A_w = A_e + A_{ww} + A_{Nw} + A_{Sw} + A_{Fw} + A_{Bw}$$

$$S_w = \rho_w^0 u_w^0 \frac{\Delta x \Delta y \Delta z}{\Delta t} - (P_P - P_W)\Delta y \Delta z$$

$$+ (u_P - u_{ww})\left(\frac{1}{Re_t}\right)_P \frac{\Delta y \Delta z}{\Delta x} - (u_W - u_{ww})\left(\frac{1}{Re_t}\right)_{ww} \frac{\Delta y \Delta z}{\Delta x}$$

$$+ \left[(v_{Nw} - v_{Nww})\left(\frac{1}{Re_t}\right)_{nw} - (v_w - v_{ww})\left(\frac{1}{Re_t}\right)_{sw}\right]\Delta z$$

$$+ \left[(w_{wf} - w_{wwf})\left(\frac{1}{Re_t}\right)_{wf} - (w_w - w_{ww})\left(\frac{1}{Re_t}\right)_{bw}\right]\Delta y$$

and

$$\rho_w^0 = \frac{\rho_P^0 + \rho_W^0}{2}$$

$$u_P = \frac{u_e + u_w}{2}$$

$$u_W = \frac{u_w + u_{ww}}{2}$$

$$u_{nw} = \frac{u_{Nw} + u_w}{2}$$

$$u_{sw} = \frac{u_{Sw} + u_w}{2}$$

$$u_{fw} = \frac{u_{Fw} + u_w}{2}$$

$$u_{bw} = \frac{u_{Bw} + u_w}{2}$$

$$G_P = \rho_P u_P$$

$$G_W = \rho_W u_W$$

$$G_{nw} = \frac{\left[\frac{\rho_N + \rho_P}{2} v_n + \frac{\rho_{NW} + \rho_W}{2} v_{nw}\right]}{2}$$

$$G_{sw} = \frac{\left[\frac{\rho^P + \rho_S}{2} v_s + \frac{\rho_W + \rho_{SW}}{2} v_{sw} \right]}{2}$$

$$G_{fw} = \frac{\left[\frac{\rho_F + \rho_P}{2} w_f + \frac{\rho_{FW} + \rho_W}{2} w_{fw} \right]}{2}$$

$$G_{bw} = \frac{\left[\frac{\rho_P + \rho_B}{2} w_b + \frac{\rho_W + \rho_{BW}}{2} w_{bw} \right]}{2}$$

Development of the equations for y and z-momentum proceed in a similar fashion and will not be repeated here in the interest of brevity.

D. PRESSURE CORRECTIONS.

1. Global Pressure Corrections.

As described by Nies [Ref. 20, pp. 50-52] and McCarthy [Ref. 23, pp.47-48], in a fixed mass - fixed volume system, overall pressure depends on the net energy added or removed from the system. Nicolette, *et al*, [Ref. 14] demonstrated that in such a system, with a uniform grid,the sum of the product of density times volume for all of the cells remains fixed at the total mass of the system. Therefore, at any time during the fire, the total mass of the system must equal the initial mass at the equilibrium density which existed before the fire was ignited. This may be expressed as:

$$\sum_i \rho_i^n (\Delta x \Delta y \Delta z)_i = \sum_i \rho_{EQ,i} (\Delta x \Delta y \Delta z)_i$$

Since the grid is uniform, the term $(\Delta x \Delta y \Delta z)_i$ is a constant, independent of time, it may be divided out of both sides of the equation leaving:

$$\sum_i \rho_i^n = \sum_i \rho_{EQ,i}$$

Now, since we are operating under the assumption that the fluid inside our burn chamber is an ideal gas, and recalling that we are also working in a fixed volume environment, the density (ρ) may now be expressed as a function of pressure and temperature only.

$$\rho_i = f(P,T)_i$$

The actual pressures and temperatures may now be expressed as the sum of an estimated value ($P^{\times}$ and $T^{\times}$) and a global correction (P_g and T_g).

$$P = P^{\times} + P_g$$

$$T = T^{\times} + T_g$$

Now applying the ideal gas law and substituting into the summation relation shown above, we can solve for the global pressure correction as

$$P_g = \frac{\sum_i P_{EQ}\left(\frac{1}{T_i} - \frac{1}{T^{\times}}\right) - \sum_i \frac{P^{\times}}{T^{\times}}}{\sum_i \frac{1}{T^{\times}}}$$

This relation is then iterated until a global pressure correction is obtained which conserves mass for all the cells.

2. Local Pressure Corrections.

As explained by both McCarthy [Ref. 23, pp.49-51] and Nies [Ref. 20, pp. 52-54], the method for obtaining the local pressure correction was developed by Patankar [Ref. 26, pp. 120-126] and Doria [Ref. 18, pp. 26-32] and is similar to the method used for determination of the global pressure correction. For this correction, the pressure distribution found during the previous time step is used to estimate the velocity field. Continuity is then applied and residual mass terms (S_{mp}) are calculated for each cell. Based on these residual mass terms a pressure correction is estimated and the process is repeated until the values of S_{mp} fall below a previously established threshold value at which point the final value of the correction is now known. As in the determination of the global pressure correction, the total pressure is expressed as the sum of the estimated pressure and the local correction.

$$P = P^{\times} + P'$$

where P is the total pressure, $P^{\times}$ is the estimated pressure, and P' is the local pressure correction. The local pressure correction may now be expressed in its finite difference form

$$A_P P'_P = A_E P'_E + A_W P'_W + A_N P'_N + A_S P'_S + A_F P'_F + A_B P'_B - S_{mp}\Delta x\Delta y\Delta z$$

where

$$A_E = \frac{\rho_e(\Delta y\Delta z)^2}{A_e + \rho_e \dfrac{\Delta x\Delta y\Delta z}{\Delta t}}$$

$$A_W = \frac{\rho_w(\Delta y\Delta z)^2}{A_w + \rho_w \dfrac{\Delta x\Delta y\Delta z}{\Delta t}}$$

$$A_N = \frac{\rho_n(\Delta z\Delta x)^2}{A_n + \rho_n \dfrac{\Delta x\Delta y\Delta z}{\Delta t}}$$

$$A_S = \frac{\rho_s(\Delta z\Delta x)^2}{A_s + \rho_s \dfrac{\Delta x\Delta y\Delta z}{\Delta t}}$$

$$A_F = \frac{\rho_f(\Delta x\Delta y)^2}{A_f + \rho_f \dfrac{\Delta x\Delta y\Delta z}{\Delta t}}$$

$$A_B = \frac{\rho_b(\Delta x\Delta y)^2}{A_b + \rho_b \dfrac{\Delta x\Delta y\Delta z}{\Delta t}}$$

$$A_P = A_E + A_W + A_N + A_S + A_F + A_B$$

At solid boundaries where the mass flux (G) is zero, the appropriate coefficient (A) corresponding to that boundary is also set equal to zero.

After the local pressure correction (P') is determined, new velocities are determined from the following relations:

$$u = u^{\times} + u'$$

$$v = v^{\times} + v'$$

$$w = w^{\times} + w'$$

where

$$u' = \frac{(P'_P - P'_W)\Delta y \Delta z}{A_w + \rho_w \dfrac{\Delta x \Delta y \Delta z}{\Delta t}}$$

$$v' = \frac{(P'_P - P'_S)\Delta z \Delta x}{A_s + \rho_s \dfrac{\Delta x \Delta y \Delta z}{\Delta t}}$$

$$w' = \frac{(P'_P - P'_B)\Delta x \Delta y}{A_b + \rho_b \dfrac{\Delta x \Delta y \Delta z}{\Delta t}}$$

The residual mass (S_{mp}) is again compared to the threshold value and the entire process is iterated if necessary.

E. INITIAL AND BOUNDARY CONDITIONS.

Before this system of equations can be solved, appropriate initial and boundary conditions must be determined and applied. As in the previous work conducted by Nies, Raycraft, Houck and McCarthy [Refs. 20 , 21, 22, and 23], these conditions are established as follows:

1. Initial Conditions.

The initial conditions for the model are determined by the conditions existing inside the test chamber just prior to starting the fire. It is assumed that the air is uniformly at rest, thus all components of velocity are set equal to zero throughout the chamber. The temperature inside the chamber is assumed to be uniform and equal to the ambient temperature outside the chamber, therefore the non-dimensional temperature field is set equal to 1.0 throughout the model. Finally, pressure and density are also assumed to be uniformly distributed and in static equilibrium.

2. Boundary Conditions.

The chamber walls are constructed of standard 3/8 inch sheet steel [Ref. 2, p. 10], therefore they are presumed to be non-porous. This allows all velocity components

to be set equal to zero at the wall (the so called "no slip" condition), and the mass flux across the wall is also set to zero (the "impermeable wall" condition). Since there is no heat generation inside the walls of the chamber, it can be assumed that no discontinuities exist between the temperature of the surface of the wall and the air immediately adjacent to it. Therefore the inside surface temperature of the wall must be identically equal to the temperature of the fluid immediately adjacent to it. These condition may be expressed as follows:

$$u_{surf} = 0$$

$$v_{surf} = 0$$

$$w_{surf} = 0$$

$$T_{surf} = T_{air}$$

Finally, conservation of energy must also be satisfied at the wall. Therefore

$$q_r - k_{air} \frac{\delta T_{air}}{\delta n} = - k_{surf} \frac{\delta T_{surf}}{\delta n}$$

where n represents the inward pointing normal at that location; and q_r represents the energy transfer due to thermal radiation.

F. MODELING OF PHYSICAL PHENOMENA.

1. Wall Conduction.

Heat losses through the walls are calculated assuming one dimensional, unsteady heat conduction through walls of uniform conductivity. A constant convective heat transfer condition is assumed to exist between the exterior of the wall and the environment.

2. Turbulence.

As explained by McCarthy [Ref. 23, pp. 15-16] and Nies [Ref. 20, pp. 39-40] the turbulence model used in this code is a simple algebraic model first developed by Nee and Liu [Ref. 28]. This model calculates an effective viscosity (μ_{eff}) and an effective conductivity (k_{eff}) for recirculating buoyant flows with widely fluctuating turbulence levels. Development of these equations, as used in this model, is shown in Nies [Ref. 20, p. 39-40].

3. Radiation.

The radiation model used in this code considers only surface radiation while considering the gas and smoke to be transparent. Developed and explained in detail by Raycraft [Ref. 21], the model is based on the net radiosity method discussed by Sparrow and Cess [Ref. 29]. As summarized by McCarthy [Ref. 23, p. 17], the model treats both the chamber walls and the flame areas as grey, diffuse surfaces.

III. THE COMPUTER PROGRAMS

A. INTRODUCTION.

Running the computer code for this project, from initial data input to final graphical output, is a three step process. Initial input is accomplished using the program FIREBLD; numerical data output is generated by the program FIRE; and final graphical output is provided by the programs ISOTHERM and VELOCITY, both of which are written to utilize the graphics software developed by the National Center for Atmospheric Research (NCAR).

B. PROGRAM FIREBLD.

Program FIREBLD [Appendix A] is used to build and or modify the data file FIRE.DATA which provides the input data required to run the main code, FIRE. Originally part of SUBROUTINE INPUT in FIRE, FIREBLD represents an effort to improve the "user friendliness" of the code. It accomplishes this by creating an interactive input environment which attempts to minimize the user's need for detailed knowledge of the internal operation of FIRE. As the user friendliness of the SUBROUTINE INPUT increased, so did its length. It quickly became the largest subroutine in FIRE, and in an effort to streamline that code, it was converted into a separate program. This conversion had two advantageous effects:

1. It increased the run time options for FIRE by eliminating the need for direct operator input while running the program. This allows FIRE to take advantage of the time savings associated with running in background and or batch modes; and
2. Elimination of the necessity for direct operator input has reduced the total run time of the code (FIRE) by placing the slow process of data input in a separate program.

FIREBLD is designed to query the user regarding the various input parameters, indicating the proper units when appropriate. It begins by looking for a previously created input file. If one is found the user is asked if it should be used, a negative response causes program termination, while a positive response causes the data file to be read. The program then continues by displaying the existing data and asking the user if any changes are desired. If a previously created input file is not located, FIREBLD automatically enters input mode and prompts the user for the required information.

Some of the input data used in this project is shown in Table 1 on page 25 and Table 2 on page 26. Additional data which may be input at the user's discretion includes:

Thermocouple data, including number and location. The present code is limited to a maximum of 20 thermocouples, but this may be changed with only minor modifications to the code.

Mass source data, including location and size. This portion of the code was not required for this project and therefore has not been tested.

Internal solids data, including location, size, conductivity, specific heat, and fan speed. The fan speed included here is for internal ventilation contained wholly inside the test chamber. This is similar to the ventilation incorporated into the code by Houck [Ref. 22]. The problems experienced by McCarthy [Ref. 23] in association with this ventilation, have not been seen here. It should also be noted that the external chamber walls are input as internal solids.

Table 1. INPUT DATA

	Directions		
	X	Y	Z
Chamber dimensions			
Length (feet)	20.0	20.0	10.0
Wall thickness (inches)	0.375		
Floor/ceiling thickness (inches)	0.5		
Number of Computational Cells			
Inside the chamber	20	20	10
Total number of cells	24	24	14
Fire location			
Starting node	19	4	0
Ending node	20	8	1
Times (seconds)			
Total length of run	90.0		
Incremental time step	0.003125		
Between data saves	30.0		
Between data saves for plotting	30.0		
Fire start time	0.0		

Table 2. WALL DATA: The following data is input through FIREBLD to identify the location of the chamber walls.

	Location		
	X	Y	Z
Wall #1, starting node	0	0	0
Wall #1, number of nodes	1	21	11
Wall #2, starting node	0	0	0
Wall #2, number of nodes	21	1	11
Wall #3 (floor), starting node	0	0	0
Wall #3, number of nodes	21	21	1
Wall #4 (ceiling), starting node	0	0	11
Wall #4, number of nodes	21	21	11
Wall #5, starting node	0	21	0
Wall #5, number of nodes	21	1	11
Wall #6, starting node	21	0	0
Wall #6, number of nodes	1	21	11

C. PROGRAM FIRE.

Operational details of the main program FIRE [Appendix B] are provided by McCarthy [Ref. 23, pp. 55-56]. The only significant difference between his version of the program and the present version (neglecting the obvious difference of compartment geometry) is that the separate program required by McCarthy for the generation of view factors, is incorporated into the present code as a subroutine, SUBROUTINE VIEW. Also, the present code inputs the wall locations and properties as "internal solids", as noted above.

Heat input from the fire is modeled as a volumetric heat source based on the dimensions of the fire as provided by FIREBLD. The magnitude of this heat source is calculated in SUBROUTINE CALQ. It is based on the known heat of combustion of the fuel used (2600 Btu/lbm, as provided by NAWC, China Lake, CA) and an estimated burn rate of 1.0 lbm/sec. However, due to temperature and velocity instabilities associated with this high heat generation rate, it was necessary to reduce the burn rate by two

orders of magnitude to 0.01 lbm/sec in order to achieve stable output data. Increasing the burn rate to its proper level would necessitate reductions in the time step used, and require multiple data runs to confirm stable output. Due to time restrictions placed on completion of this work, those additional data runs were left for later investigators.

In making comparisons to McCarthy's work [Ref. 23, p. 55], it is interesting to note that he used the VAXSTATION 3100 which required approximately 3600 seconds of CPU time to process one second of fire time. The current work required approximately 677 seconds of CPU time per second of fire time, utilizing the AMDAHL #### mainframe processor here at Naval Postgraduate School. Even when performance is broken down further, to a CPU seconds per fire second per cell basis, the AMDAHL out performs the VAX by a 5.5 or 6 to 1 margin. Although no attempt was made to determine the source of this improvement, the majority of it can be attributed to the increased clock speed of the AMDAHL machine, which unconfirmed estimates place at approximately 4 to 1 over the VAX.

D. GRAPHICAL ANALYSIS.

The value of graphical analysis was successfully demonstrated by McCarthy [Ref. 23]. This work utilizes a different graphics software package in order to standardize the software used at both Naval Postgraduate School and the University of Notre Dame. The software package chosen, NCAR Graphics, was created at the National Center for Atmospheric Research (NCAR) and is intended specifically for scientific applications. The primary advantage of NCAR over the previously used graphics software is in the flexibility of output presentation built in to the NCAR software.

The two graphics program used in the present study, ISOTHERM [Appendix C] and VELOCITY [Appendix D] each utilize the three dimensional PLOT.DATA file created by FIRE to generate two dimensional temperature and velocity profiles as requested by the user. Each program allows the user to specify the desired two dimensional view (or section) (i.e. Plan view, X-Z profile, or Y-Z profile) and the location (or elevation) in the third dimension where that section is to be taken. Each program then performs two linear interpolations, the first to locate the requested section, and the second to convert the grid used by FIRE to a uniform grid as required by the NCAR Graphics software.

Scaling of the output in each program is accomplished in different manners. VELOCITY sets the maximum vector size in any given plot by the maximum interpolated velocity for that section (NOTE: velocities are shown in centimeters per second). While the color scale used in each plot generated by ISOTHERM is determined by dividing the

range from ambient (70 ° F = 21.1 ° C) to the maximum temperature in the non-interpolated data into 14 color zones. These scaling techniques are examples of the flexibility of the NCAR Graphics software, and each user may customize the output to satisfy the current needs/desires with minimal effort.

Color isotherm graphics are printed on a DEC LJ250 Companion Color printer using ink jet technology, attached to a VAXStation 3100 M38 standalone workstation. Velocity profiles are printed on a DEC LN03 Laser printer attached to the VAX cluster in the Mechanical Engineering Department at the Naval Postgraduate School. The following figures [Figure 8 on page 29 through Figure 25 on page 46] show the temperature and velocity profiles for each of the available view planes at 30, 60 and 90 seconds of fire time. Elevations are shown on each figure and were chosen to provide a plan view at mid-compartment height and profiles through the fire. Also included in both ISOTHERM and VELOCITY is the option to "zoom" in on any localized region of the plot as desired by the user. Figure 26 on page 47 and Figure 27 on page 48 demonstrate this capability. Note also, that the axis on all of these plots are marked (in feet) to indicate the section being viewed.

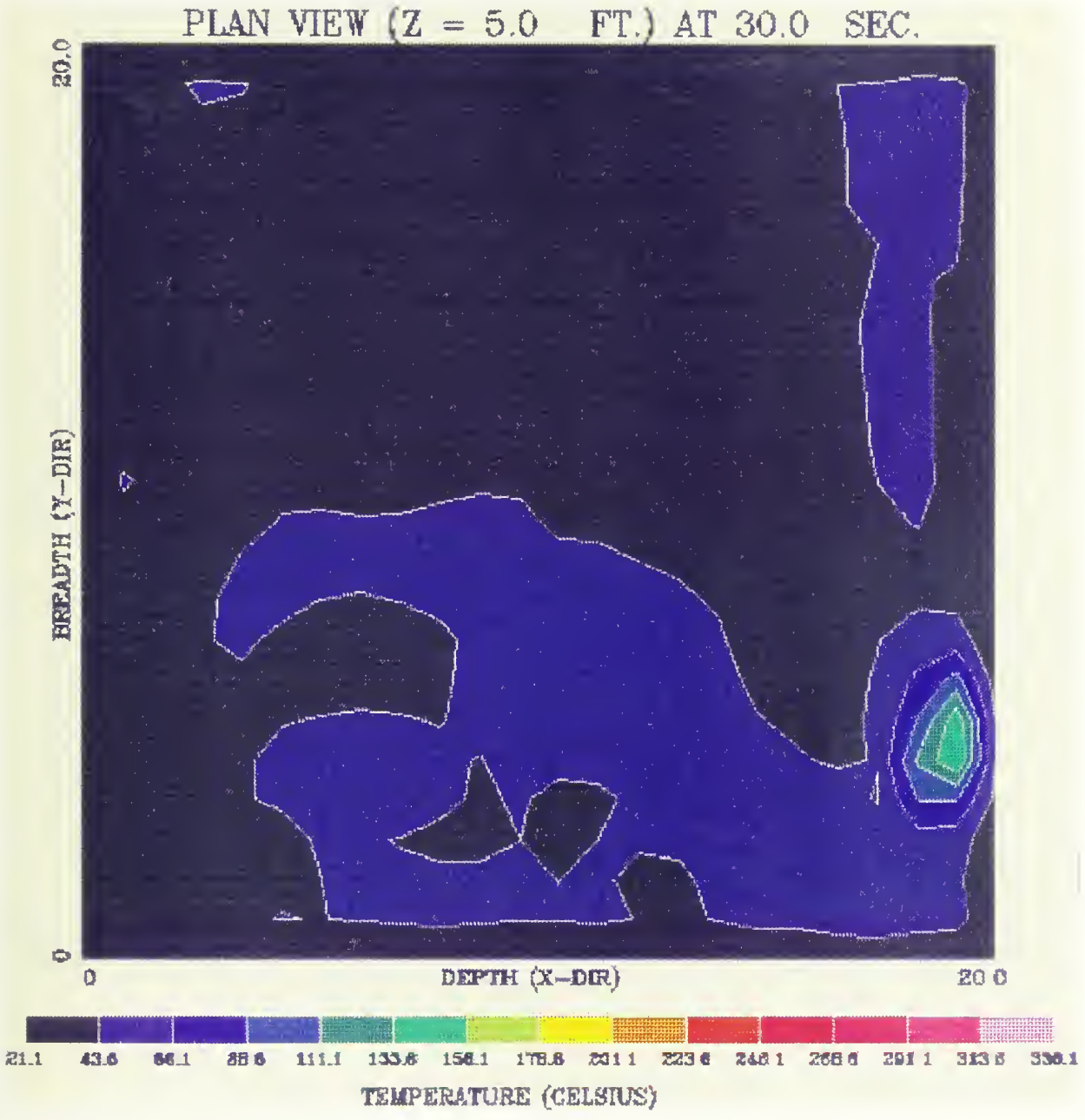

Figure 8. Temperature Profile, Plan View, 30 seconds.

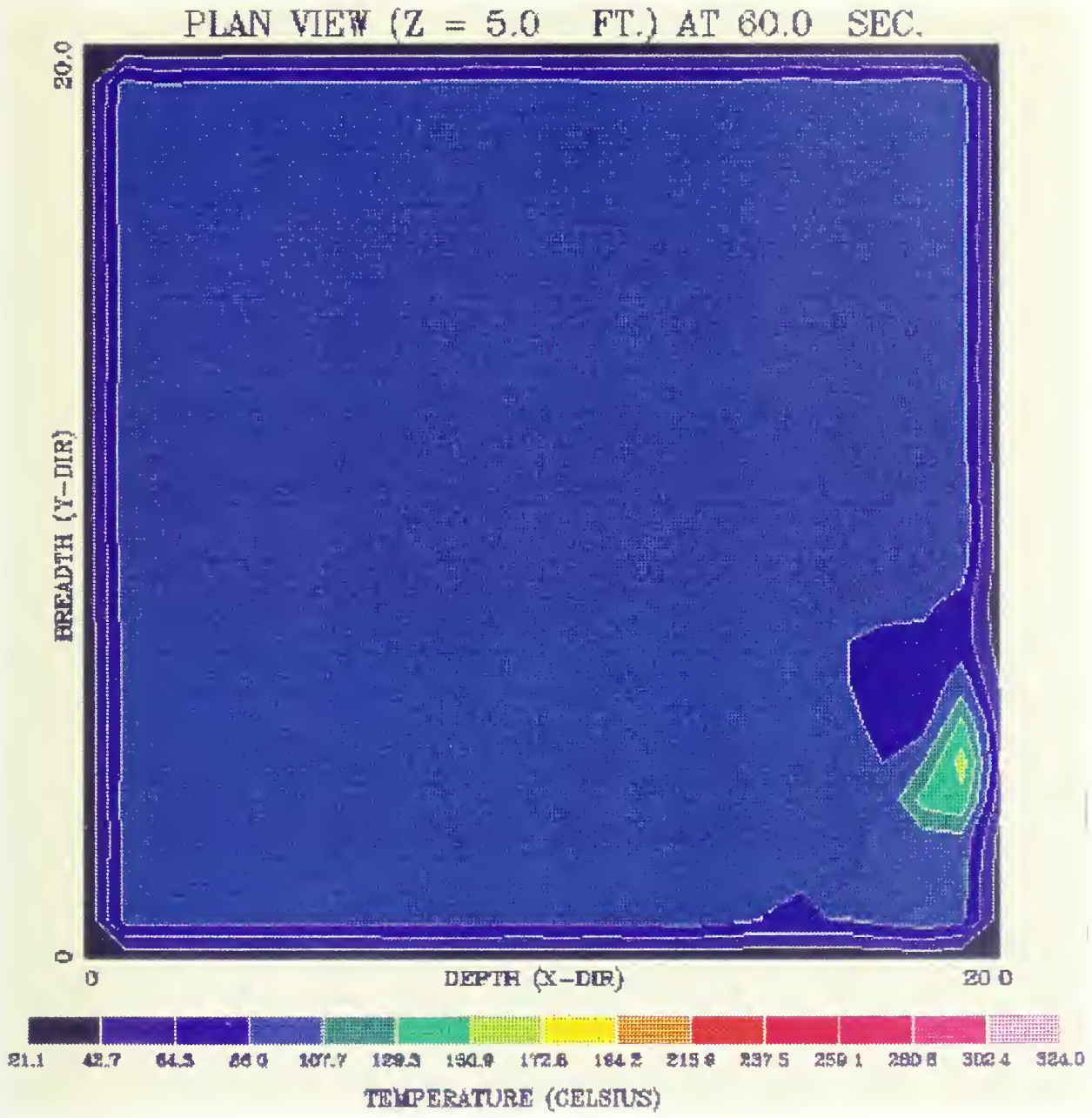

Figure 9. Temperature Profile, Plan View, 60 seconds.

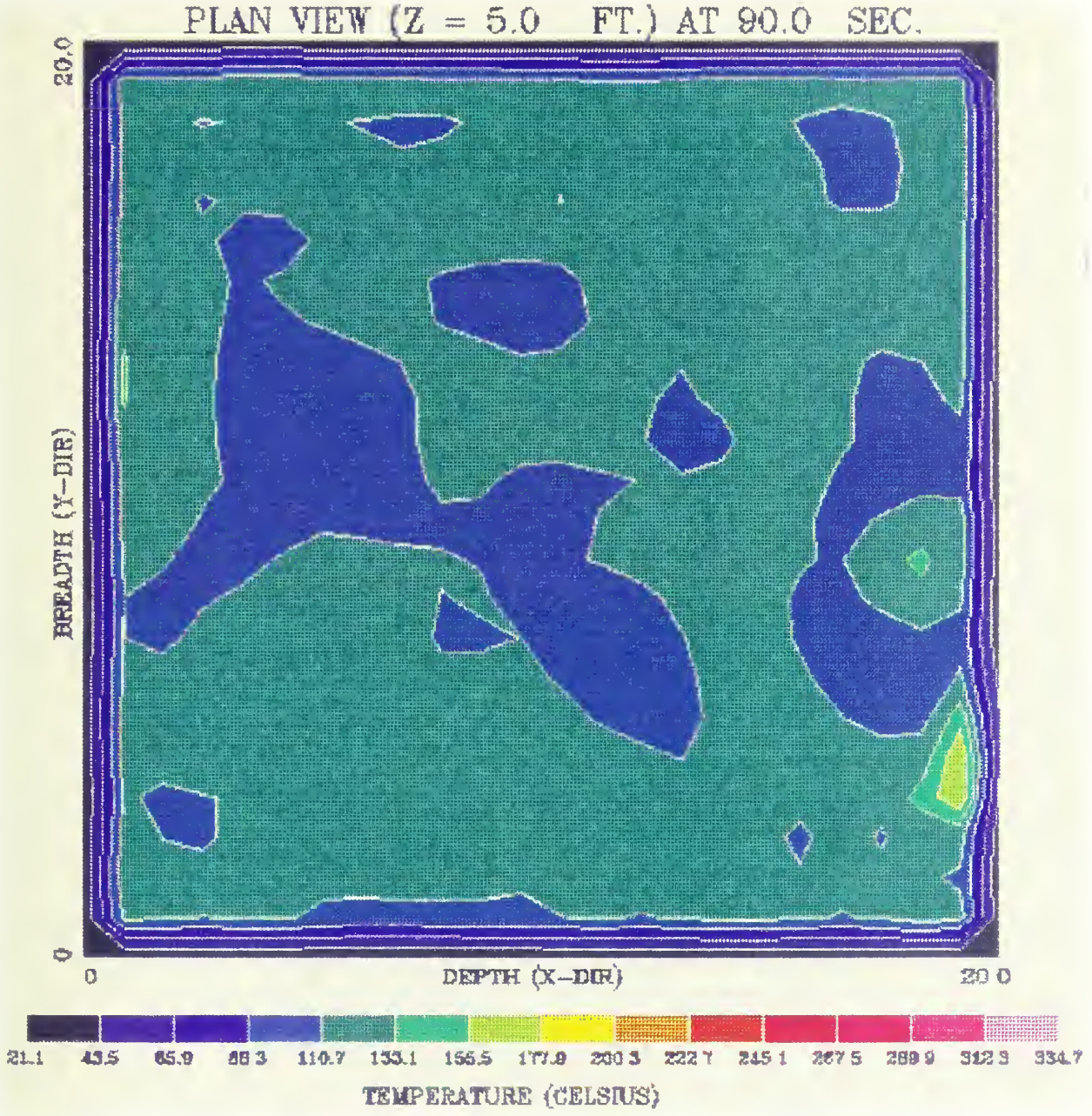

Figure 10. Temperature Profile, Plan View, 90 seconds.

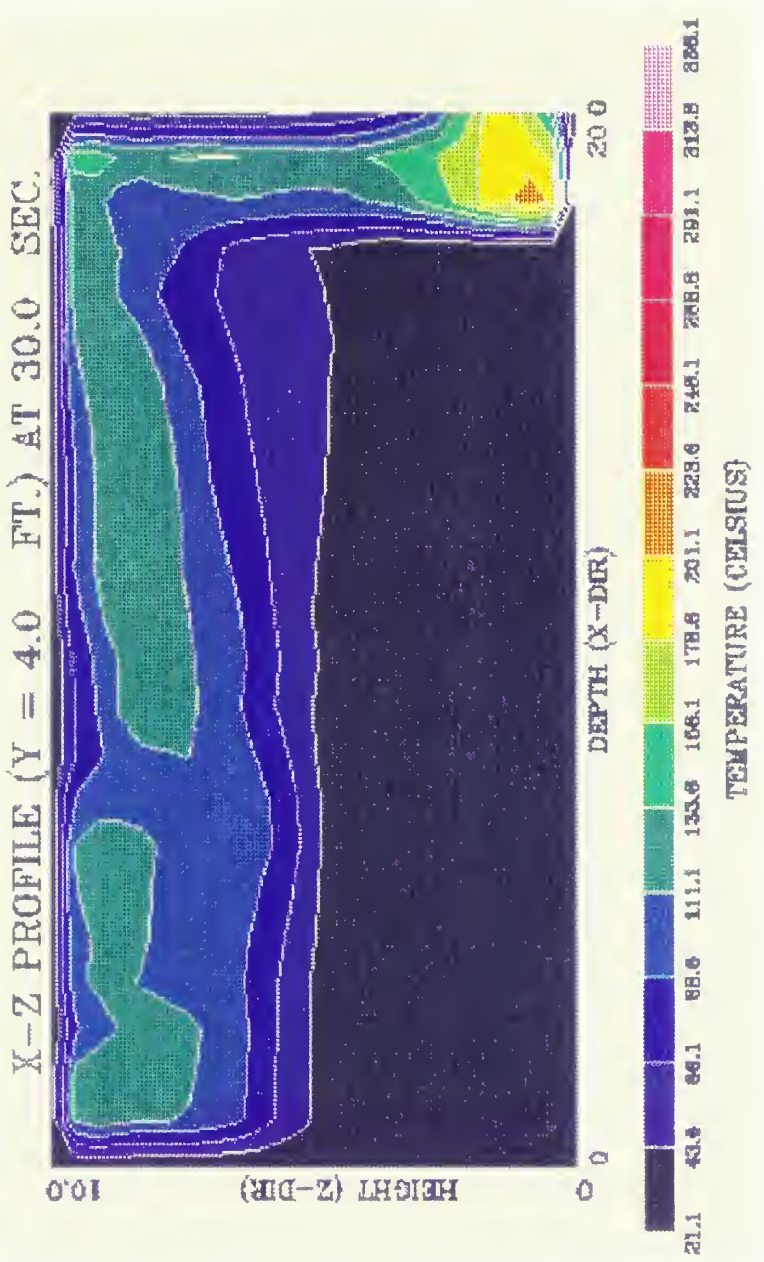

Figure 11. Temperature Profile, X-Z Profile, 30 seconds.

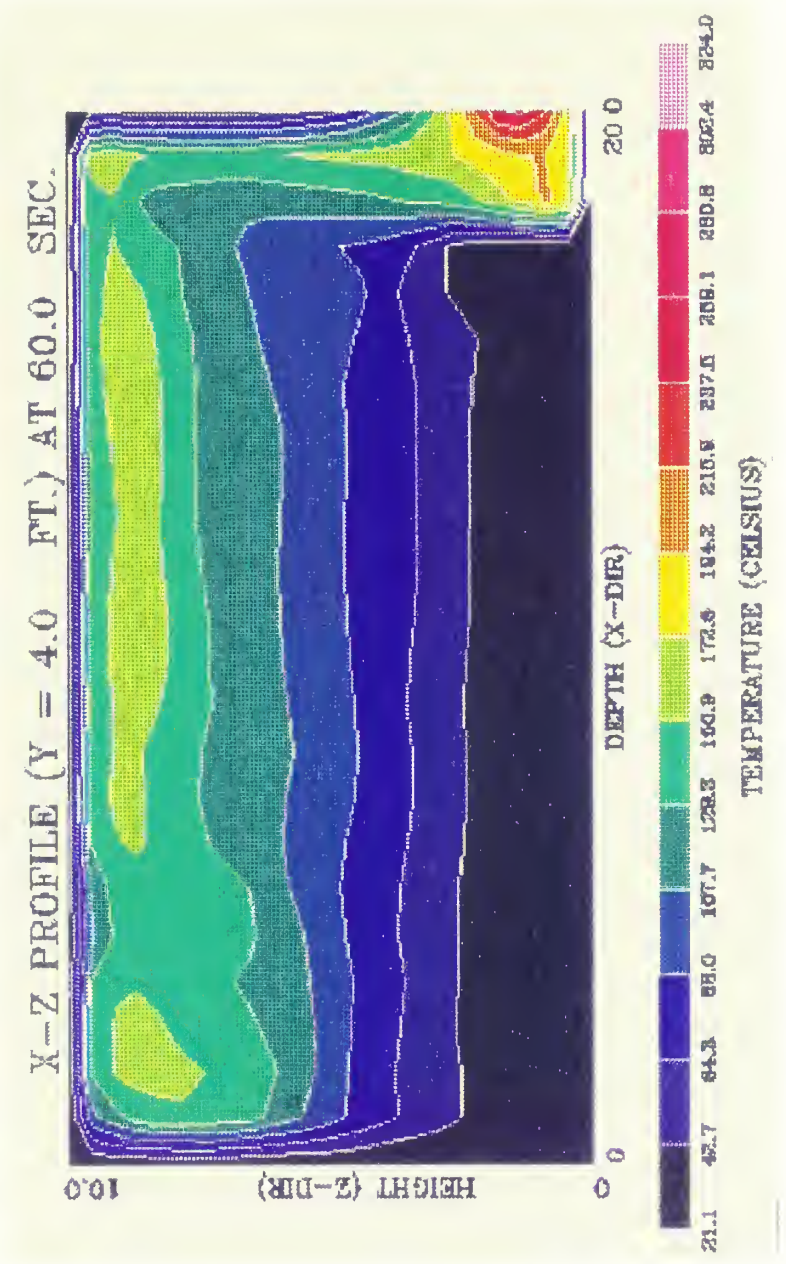

Figure 12. Temperature Profile, X-Z Profile, 60 seconds.

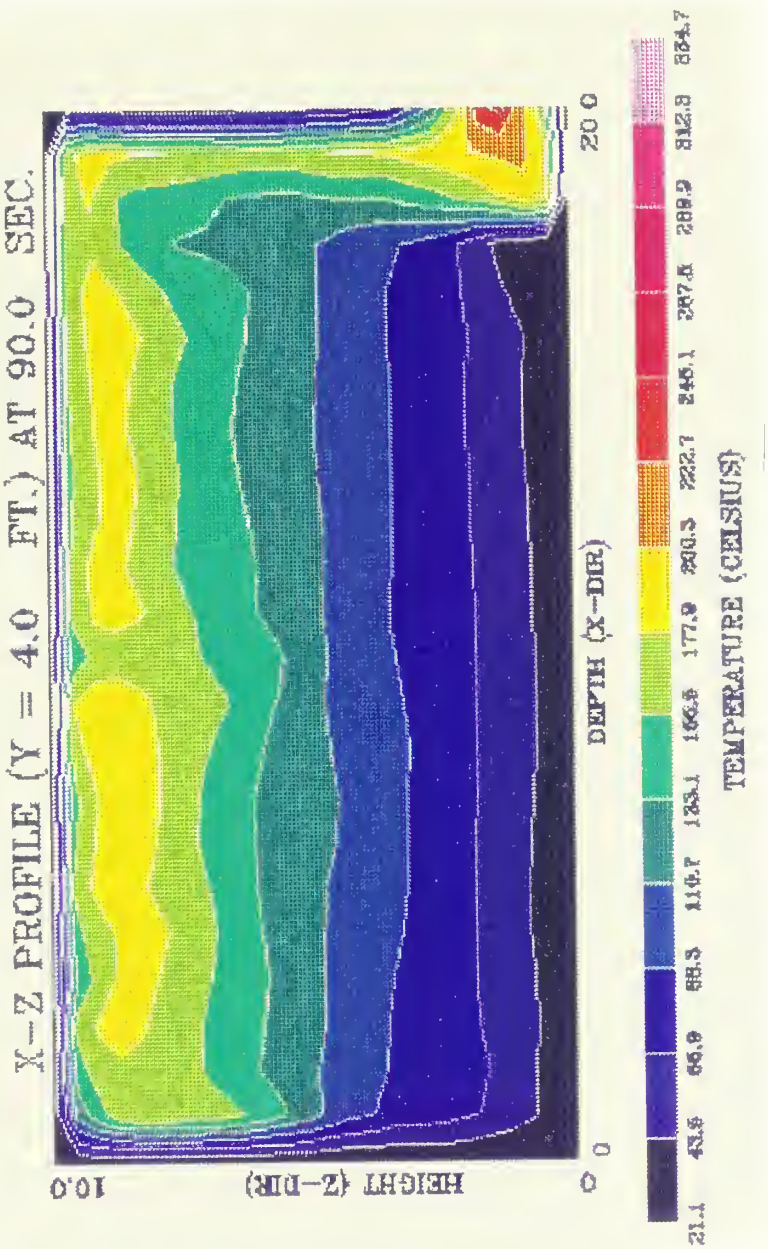

Figure 13. Temperature Profile, X-Z Profile, 90 seconds.

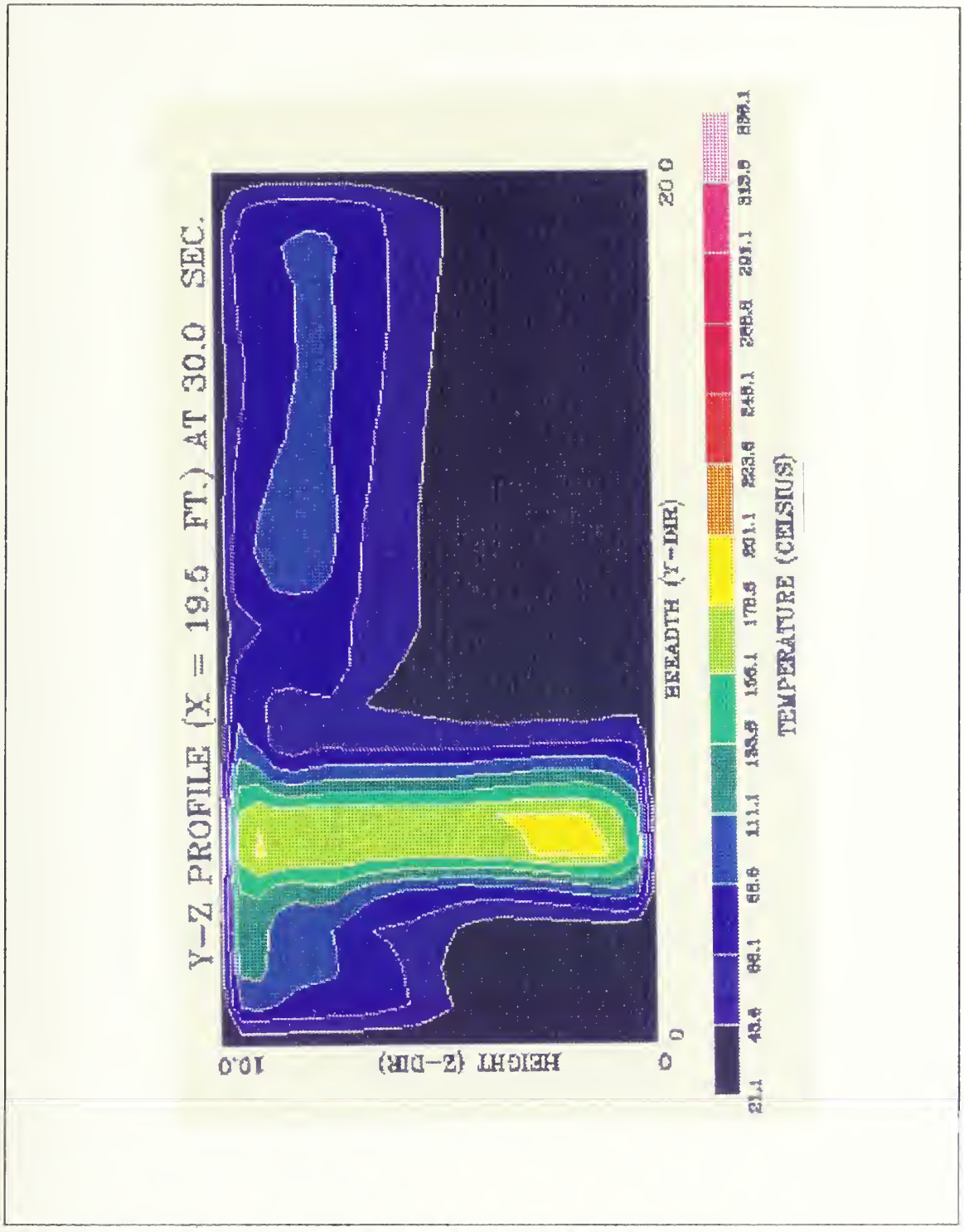

Figure 14. Temperature Profile, Y-Z Profile, 30 seconds.

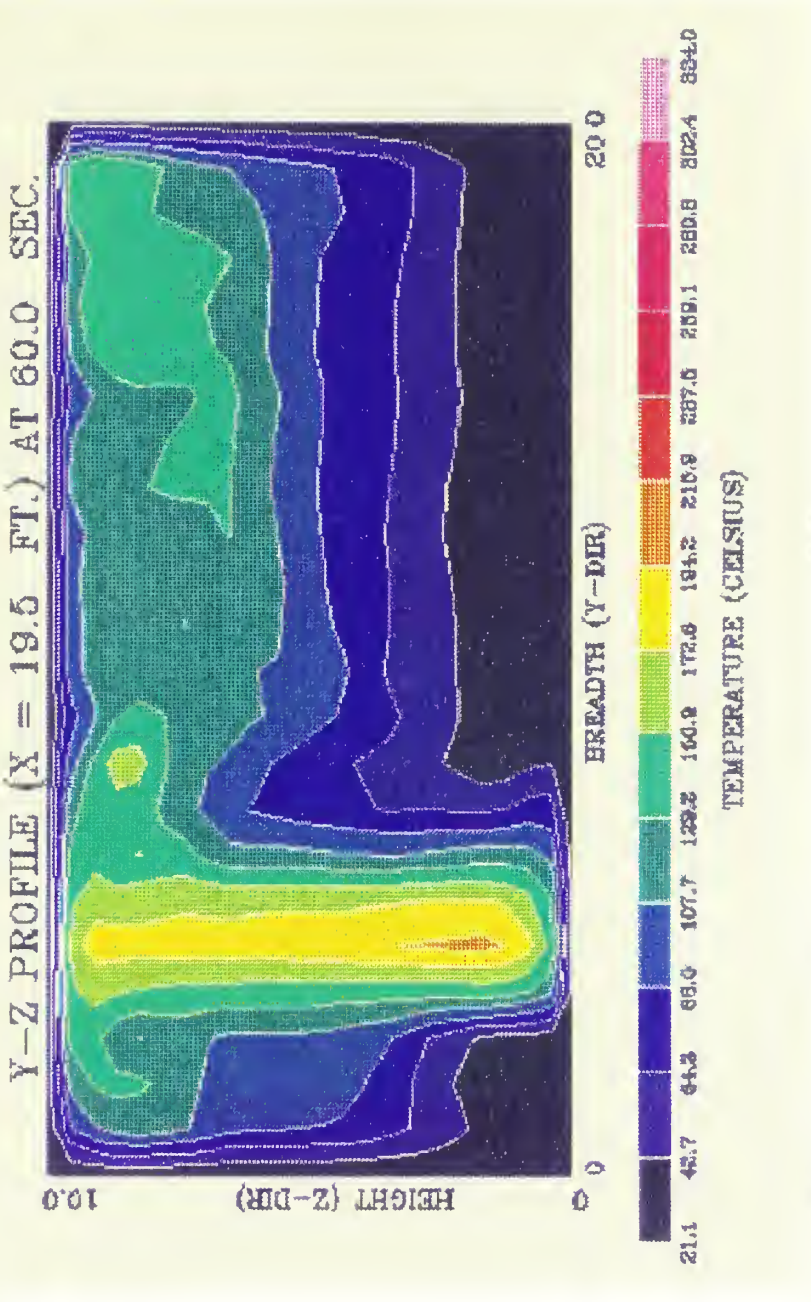

Figure 15. Temperature Profile, Y-Z Profile, 60 seconds.

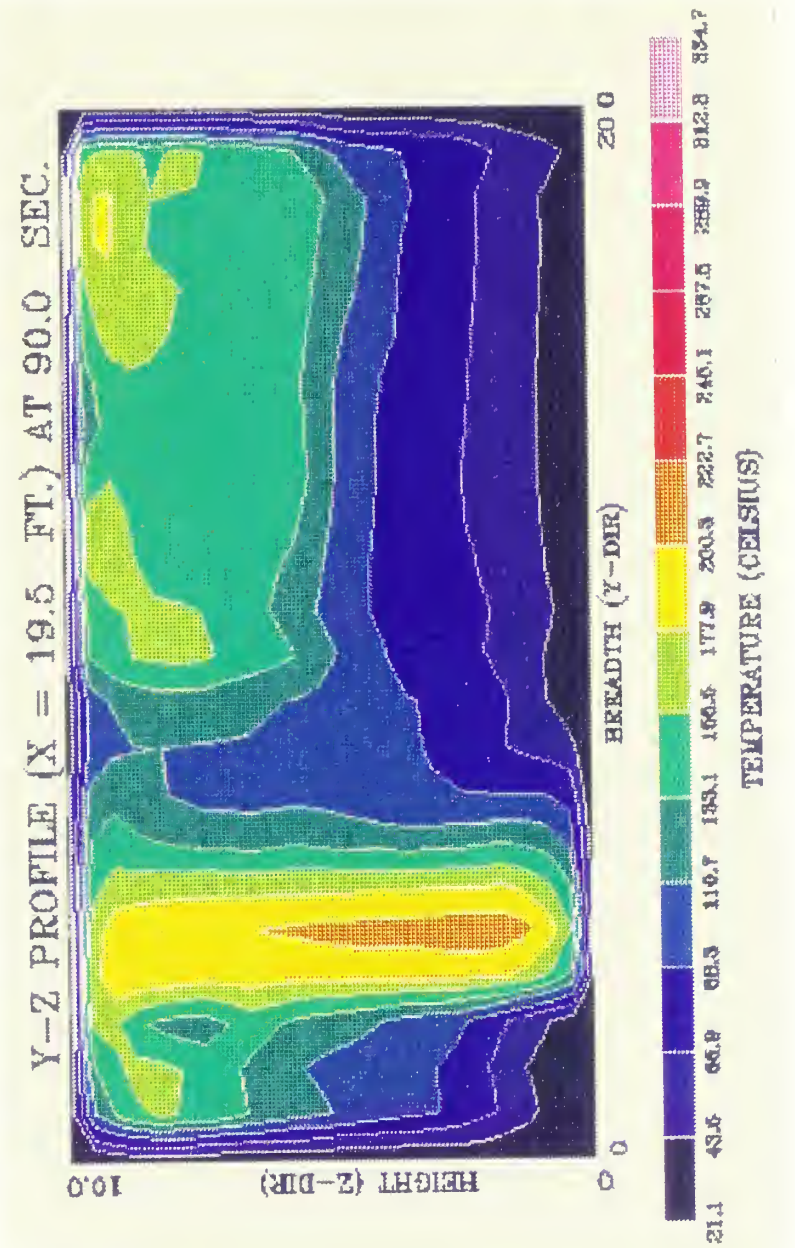

Figure 16. Temperature Profile, Y-Z Profile, 90 seconds.

THIS PAGE INTENTIONALLY LEFT BLANK

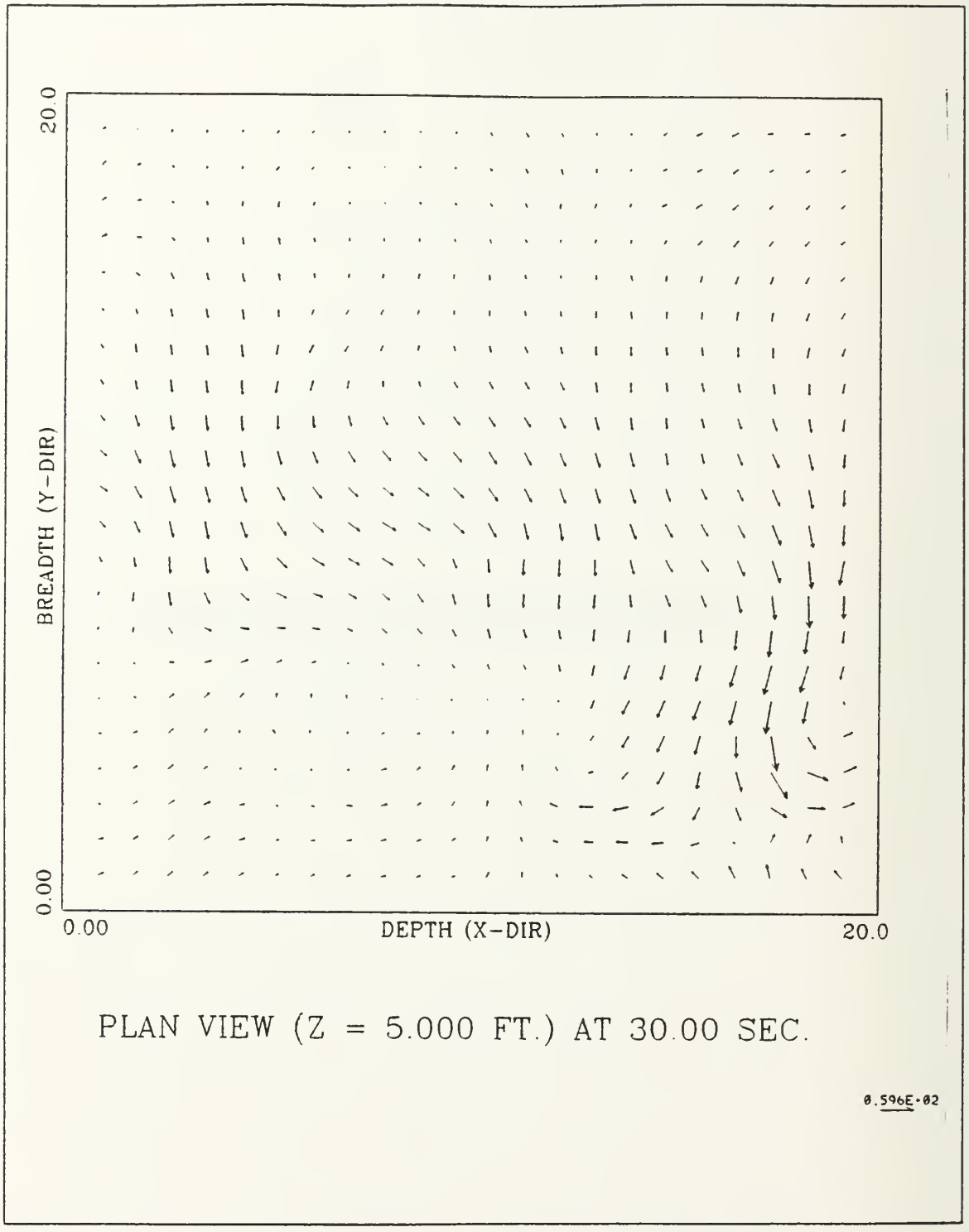

Figure 17. Velocity Profile, Plan View, 30 seconds.

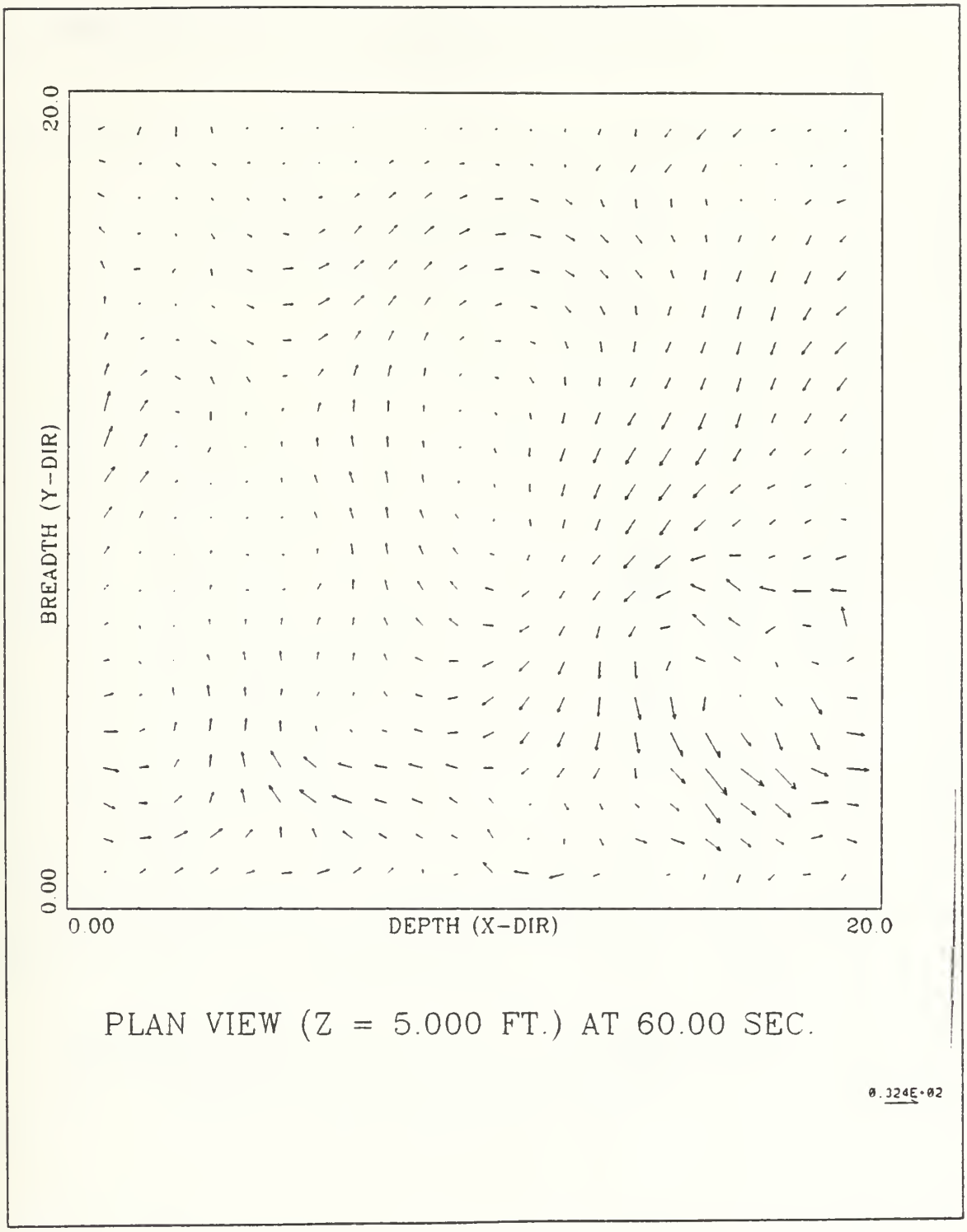

Figure 18. Velocity Profile, Plan View, 60 seconds.

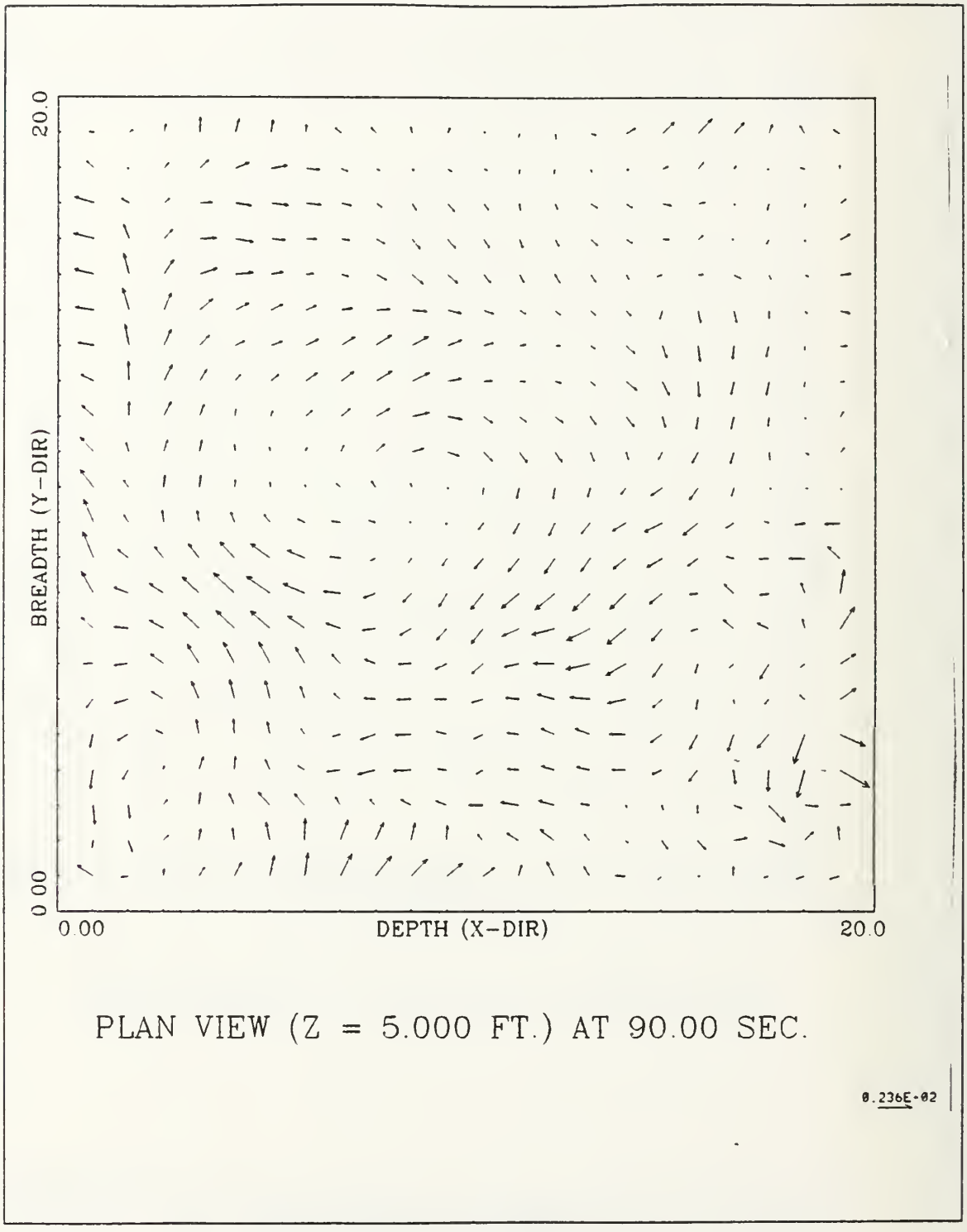

Figure 19. Velocity Profile, Plan View, 90 seconds.

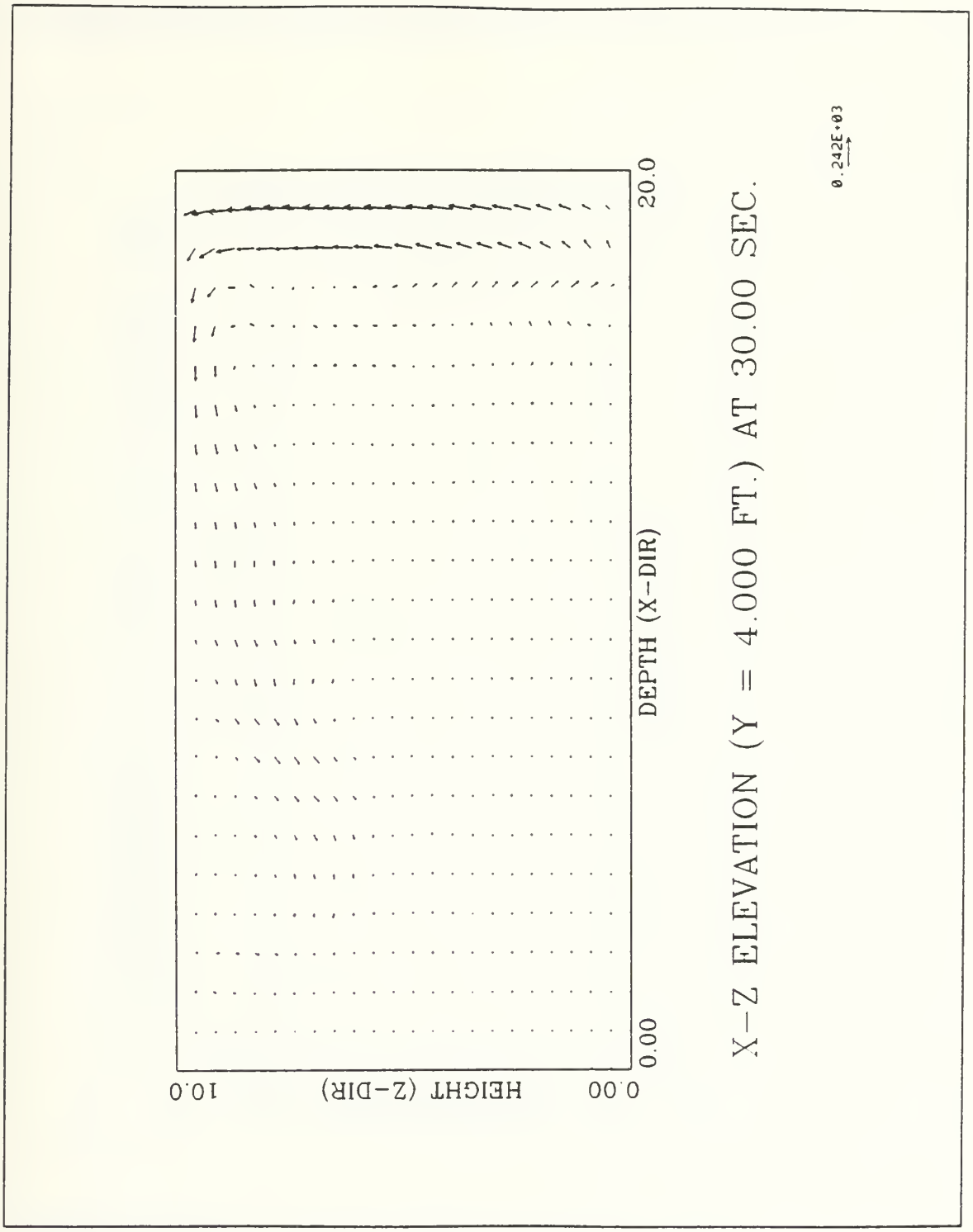

Figure 20. Velocity Profile, X-Z Profile, 30 seconds.

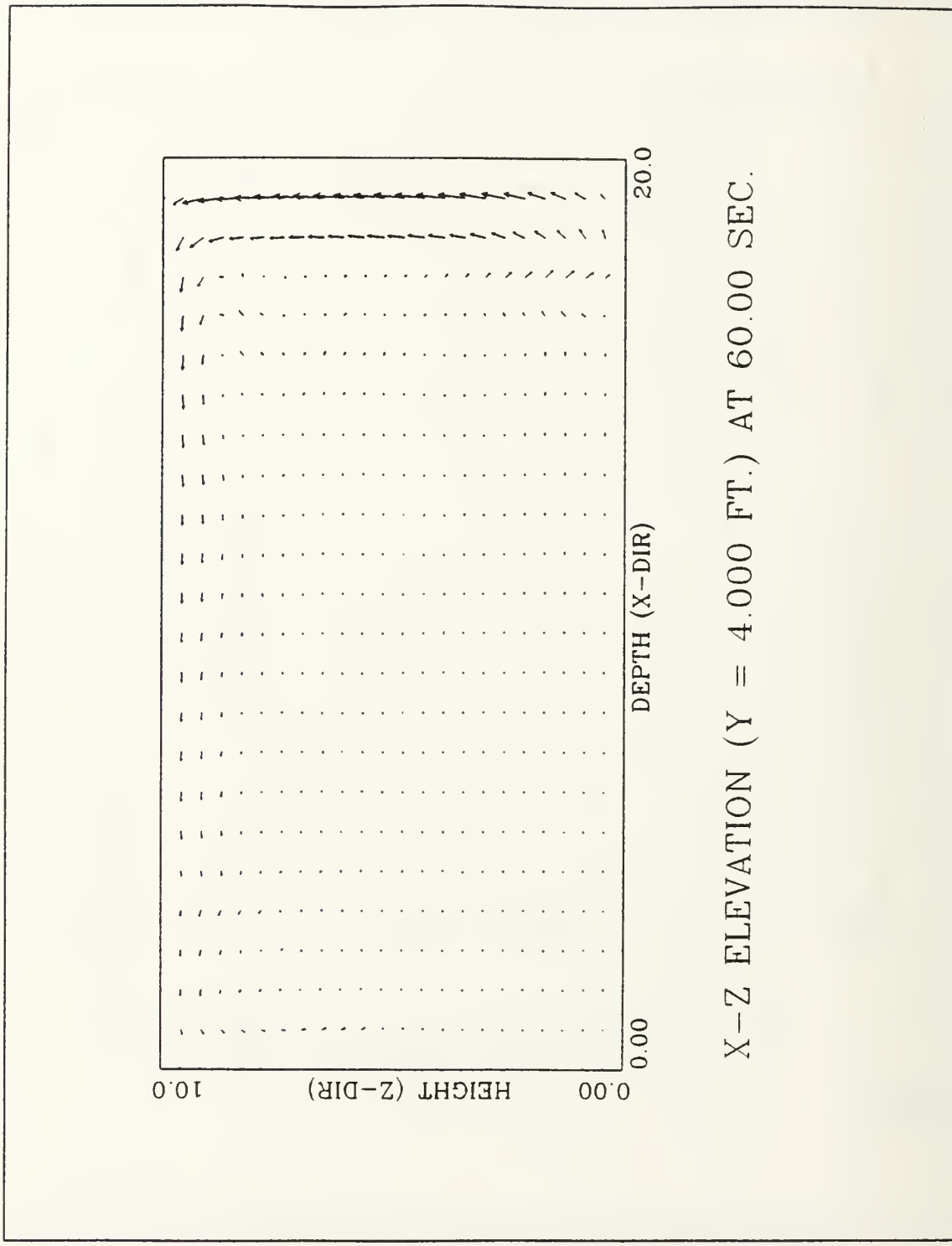

Figure 21. Velocity Profile, X-Z Profile, 60 seconds.

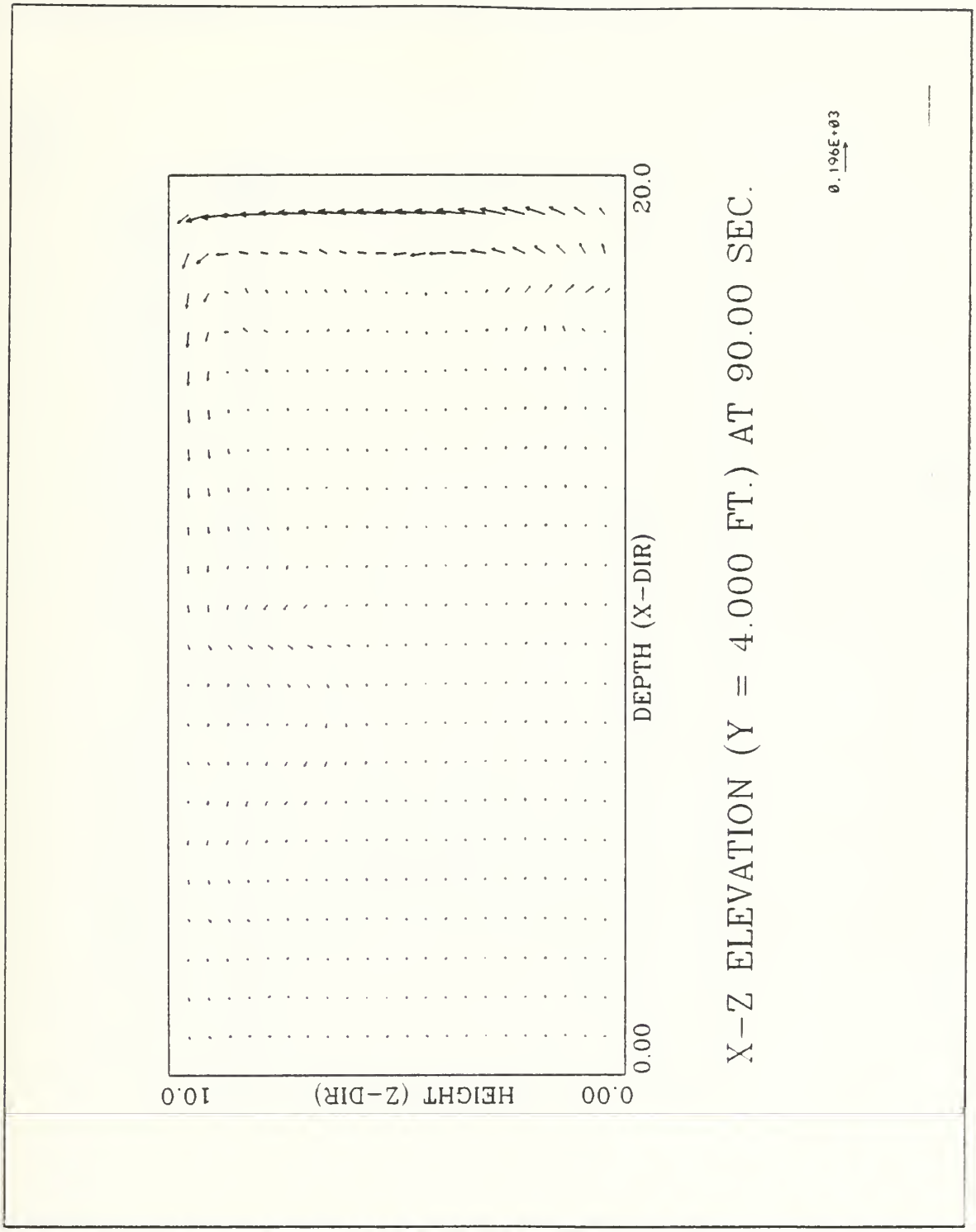

Figure 22. Velocity Profile, X-Z Profile, 90 seconds.

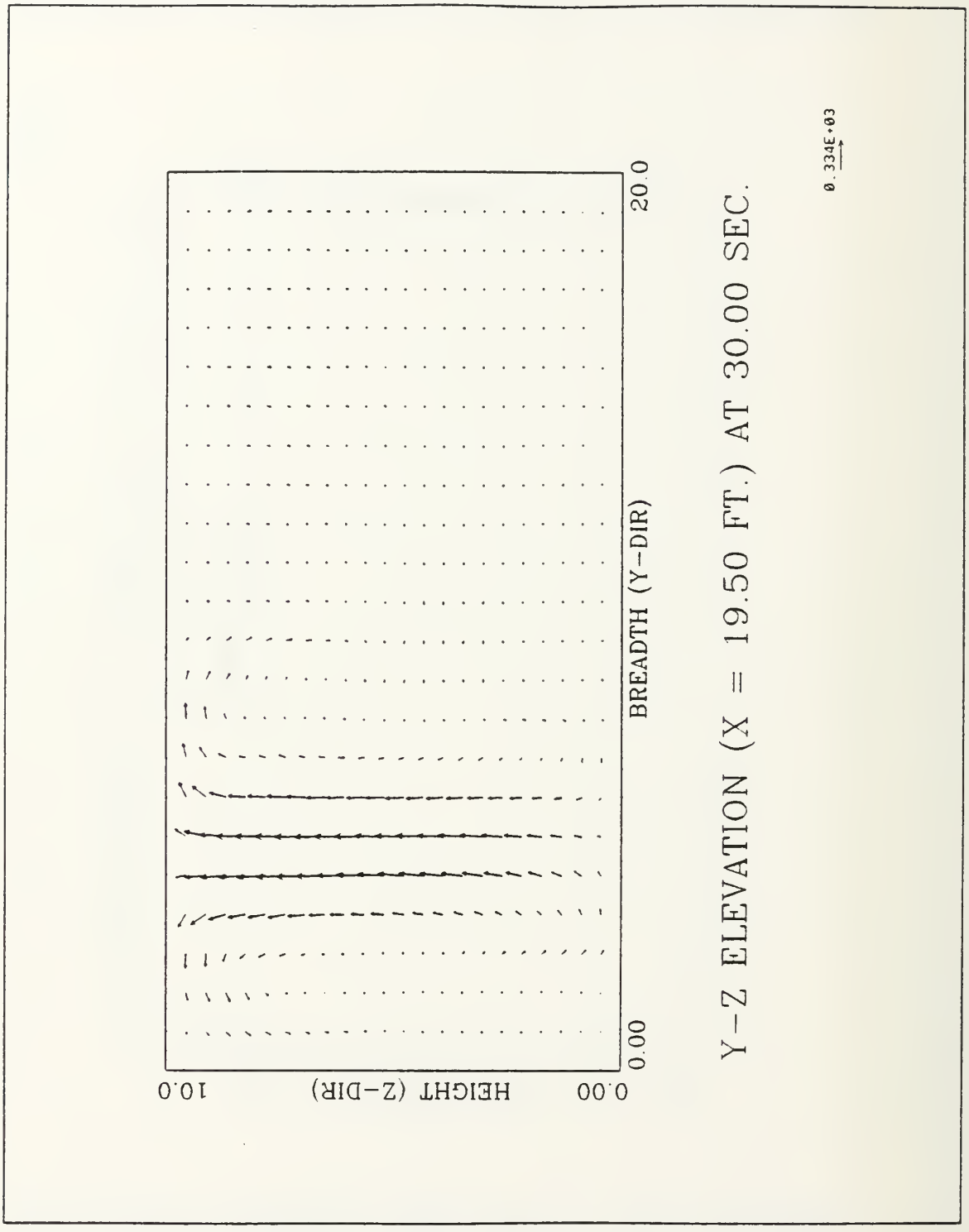

Figure 23. Velocity Profile, Y-Z Profile, 30 seconds.

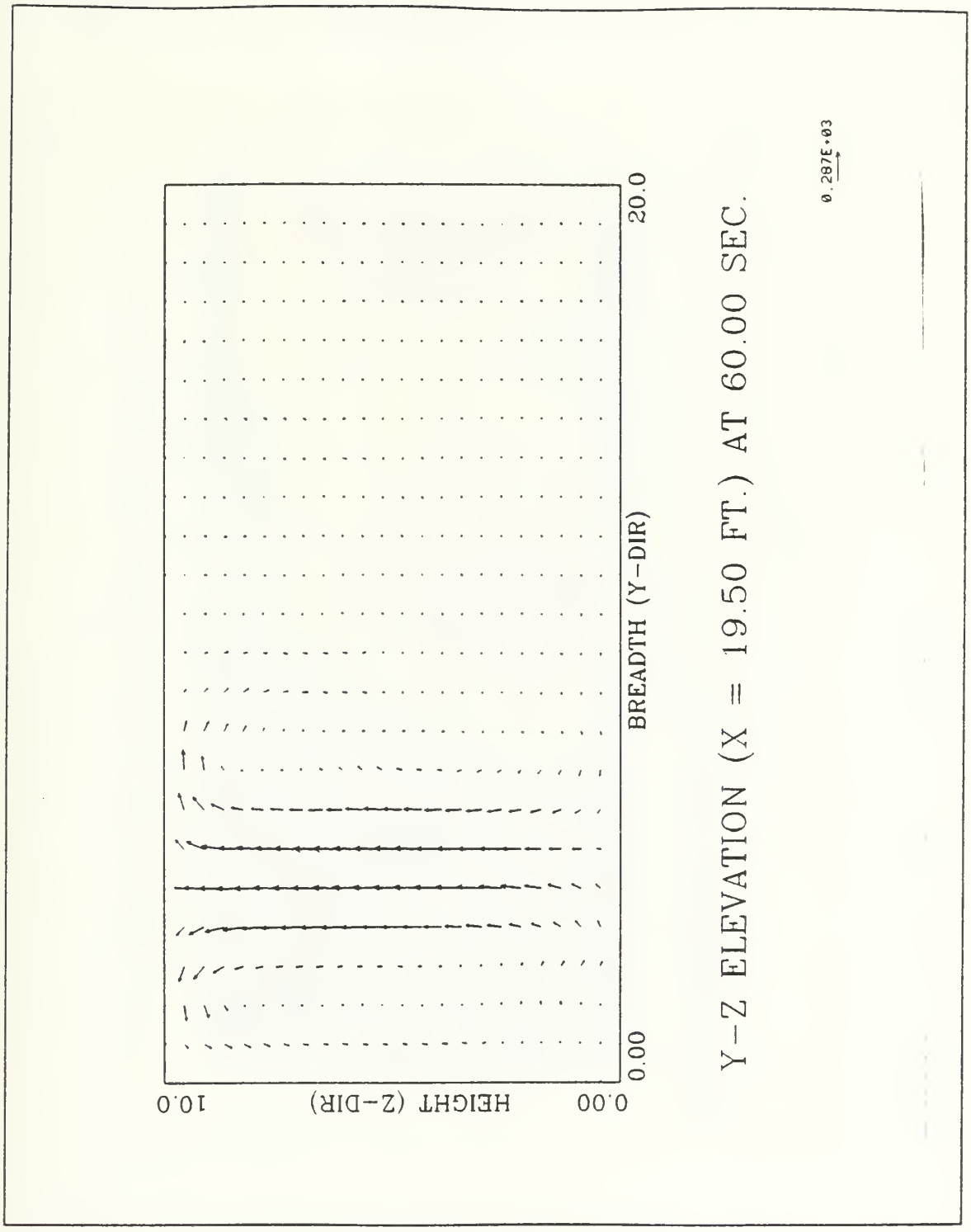

Figure 24. Velocity Profile, Y-Z Profile, 60 seconds.

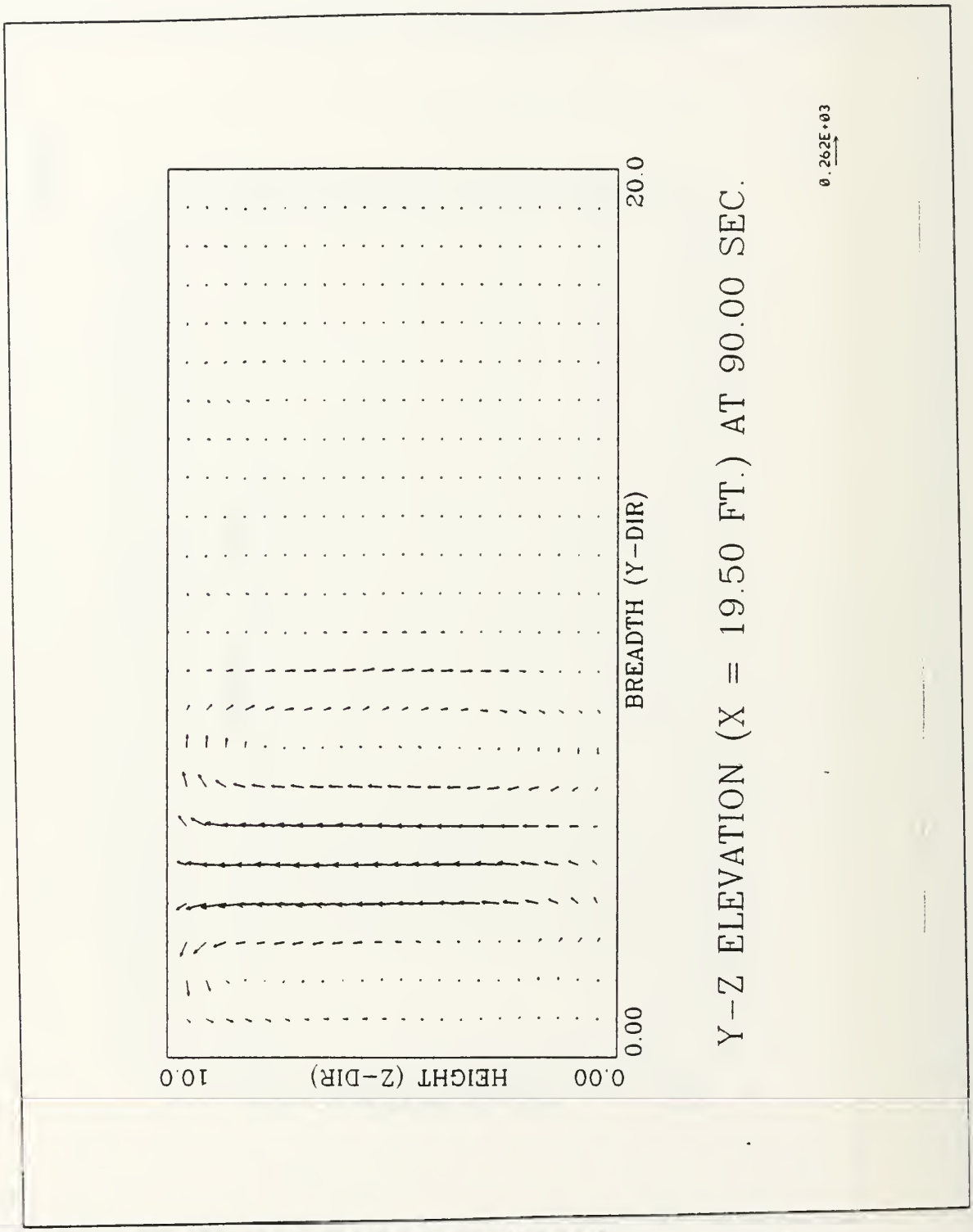

Figure 25. Velocity Profile, Y-Z Profile, 90 seconds.

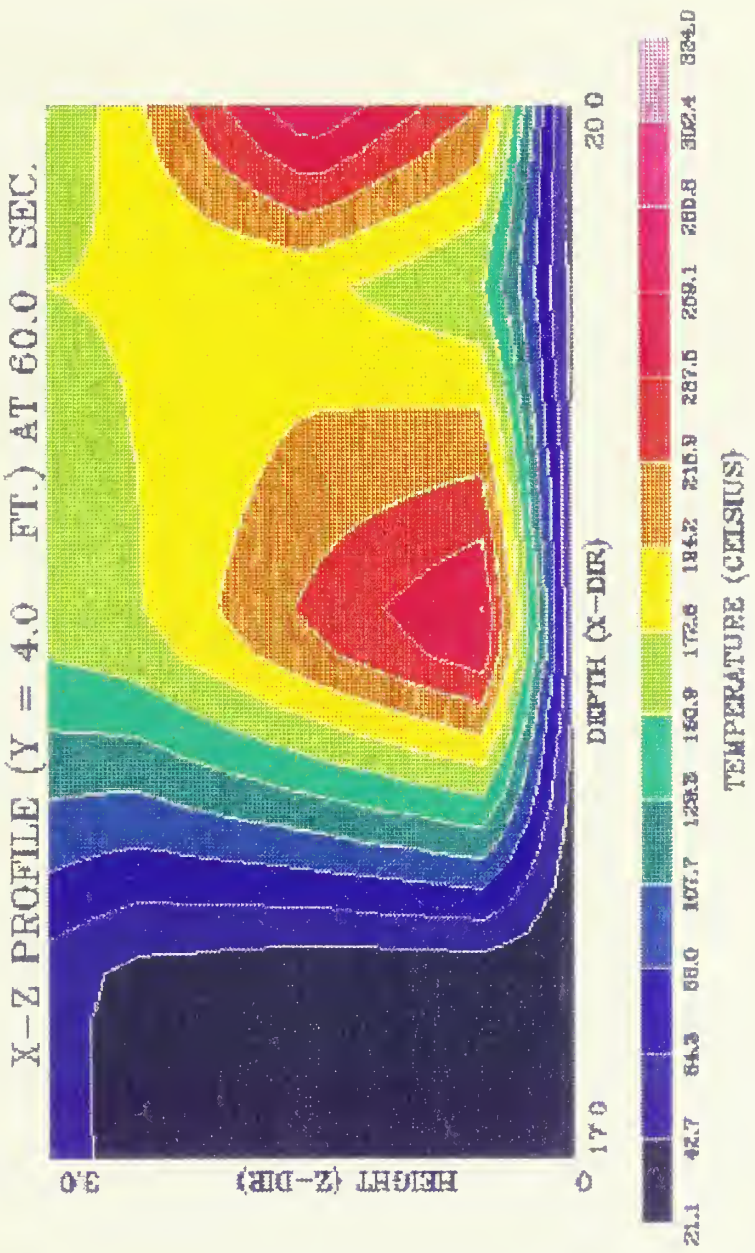

Figure 26. Example of "Zoom" feature of ISOTHERM.

THIS PAGE INTENTIONALLY LEFT BLANK

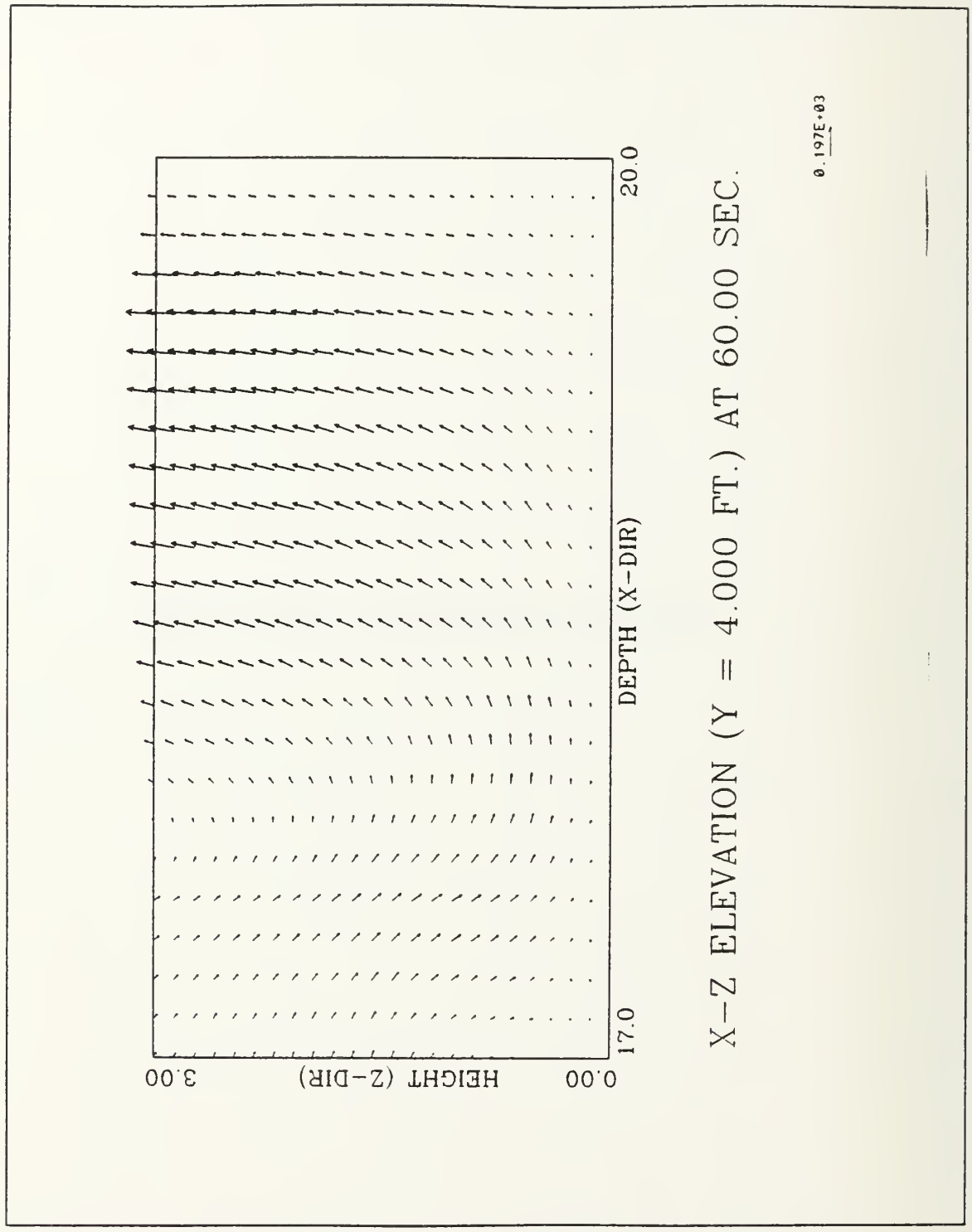

Figure 27. Example of "Zoom" feature of VELOCITY.

IV. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS.

1. Program FIREBLD represents a marked improvement in user friendly data input to FIRE.
2. Additional improvements can be made in user friendliness of FIREBLD, and thus FIRE. Specifically, all locations and dimensions should be input as length measurements (i.e. feet or meters) by the user.
3. The burn rate used by SUBROUTINE CALQ in program FIRE is artificially low and must be increased before the code can be properly validated against actual test data.
4. The NCAR Graphics software provides increased flexibility in graphics output over the previously used graphics package.
5. The temperature and velocity profiles generated by this work appear to have the expected characteristics of an actual fire. Further testing and comparison to actual test data is necessary for validation of the code.
6. The ink jet printer used to create hard copy graphical printouts of the data in this work is slow and of marginal quality for professional publication. Improved graphics printing capabilities should be pursued.

B. RECOMMENDATIONS.

1. Increase user friendliness of FIREBLD by putting all input in terms easily determined by the user. Specifically:

   Input locations in terms of length measurements vice nodal locations.

   User should have his choice of units (SI or English) when inputting data.

   Input data should not require any manipulation by the user prior to entry.
2. Increase the burn rate used by SUBROUTINE CALQ in program FIRE to realistic levels and adjust size of time step as necessary to achieve stability.
3. Expand FIRE to completely model test chamber at the Naval Air Warfare Center, China Lake, CA. Specifically, include the natural vent and adjacent compartments.
4. Validate model against actual test procedures and results from Naval Air Warfare Center, China Lake.
5. Examine alternative graphical presentation schemes available with the NCAR Graphics software. Specifically, test feasibility of combining isotherm and velocity plots into single output. Also, examine use of NCAR Graphics movie making capabilities to provide a "real time" representation of data.
6. Enhance clarity of graphics printouts by procuring access to a laser quality color printing.

APPENDIX A. PROGRAM FIREBLD

This program is used to build and/or modify the input data file required by the main program, FIRE.

```
      PROGRAM FIREBLD
C*************************************************************************
C**                                                                     **
C**              THREE-DIMENSIONAL NUMERICAL SIMULATION                 **
C**                OF A FIRE SPREAD INSIDE A BUILDING                   **
C**                                                                     **
C**                          DEVELOPED BY :                             **
C**                      H.Q. YANG AND K.T. YANG                        **
C**                                                                     **
C**        DEPARTMENT OF AEROSPACE & MECHANICAL ENGINEERING             **
C**                     UNIVERSITY OF NOTRE DAME                        **
C**                    NOTRE DAME, INDIANA, 46556                       **
C**                                                                     **
C**                             DEC. 1986                               **
C**                                                                     **
C**                            AMMENDED BY                              **
C**                         K.J. THORKILDSEN                            **
C**                    LIEUTENANT, U.S. COAST GUARD                     **
C**                                                                     **
C**                DEPARTMENT OF MECHANICAL ENGINEERING                 **
C**                     NAVAL POSTGRADUATE SCHOOL                       **
C**                    MONTEREY, CALIFORNIA  93942                      **
C**                                                                     **
C**                             SEP. 1992                               **
C**                                                                     **
C*************************************************************************
C* THIS PROGRAM BUILDS AND/OR MODIFIES THE INPUT DATA FILE REQUIRED FOR *
C* THE PROGRAM LISTED ABOVE.                                            *
C*************************************************************************
C*THIS SUBROUTINE SETS UP REQUIRED VALUES TO BEGIN THE PROGRAM.
C*
C*VARIABLES ARE:
C*  KRUN      = RESTART INDICATOR
C*  NCHIP     = NUMBER OF INTERNAL SOLID PIECES
C*  NMS       = NUMBER OF MASS SOURCES
C*  NWRP      = NUMBER OF TIME STEPS BETWEEN WRITES TO OUTPUT FILE
C*  NTHCO     = NUMBER OF THERMOCOUPLES TO PRINT OUT
C*  TMAX      = NONDIMENSIONAL MAXIMUM TIME ALLOWED
C*  XTMAX     = MAXIMUM TIME ALLOWED (SECONDS)
C*  TWRITE    = TIME BETWEEN FIELD VARIABLE OUTPUT (SECONDS)
C*  TTAPE     = TIME INTERVAL BETWEEN PLOTS (SECONDS)
C*  DTIME     = NONDIMENSIONAL TIME STEP
C*  XDTIME    = TIME STEP (SECONDS)
C*  HSTART    = FIRE START TIME (SECONDS)
C*  NHSZ(1,1) = STARTING NODE OF HEAT SOURCE, X-DIR
```

```
*  NHSZ(2,1) =                                  Y-DIR
*  NHSZ(3,1) =                                  Z-DIR
*  NHSZ(1,2) = ENDING NODE OF HEAT SOURCE, X-DIR
*  NHSZ(2,2) =                                 Y-DIR
*  NHSZ(3,2) =                                 Z-DIR
*  ICHPB     = FIRST NODE OF INTERNAL SOLID IN X DIR
*  ICHPB     = FIRST NODE OF INTERNAL SOLID IN X DIR
*  JCHPB     =                                   Y DIR
*  KCHPB     =                                   Z DIR
*  NCHPI     = NUMBER OF INTERNAL SOLID NODES IN X DIR
*  NCHPJ     =                                     Y DIR
*  NCHPK     =                                     Z DIR
*  IMSB      = FIRST MASS SOURCE NODE IN X DIR
*  JMSB      =                             Y DIR
*  KMSB      =                             Z DIR
*  NMSI      = NUMBER OF MASS SOURCE NODES IN X DIR
*  NMSJ      =                                 Y DIR
*  NMSK      =                                 Z DIR
*  RMS       = DIMENSIONLESS MASS SOURCE
*              (= CFM/(60.*H**2*U0*NMSI*NMSJ*NMSK)
*  CX,CY,CZ  = THERMOCOUPLE POSITIONS IN X,Y,Z
*************************************************************************
*DATA FILES USED IN THIS PROGRAM:
*
*       FILE # 10 = FIRE DATA B1 : INITIAL SET-UP DATA
*
*************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER ANS*1
      LOGICAL L1
      DIMENSION ICHPB(20),NCHPI(20),JCHPB(20),NCHPJ(20),KCHPB(20),
     &          NCHPK(20),CPS(20),CONS(20),WFAN(20),IMSB(20),NMSI(20),
     &          JMSB(20),NMSJ(20),KMSB(20),NMSK(20),RMS(20),CX(30),
     &          CY(30),CZ(30),NHSZ(3,2)
      PARAMETER (U0=1.0)
      INQUIRE (FILE='/FIRE DATA B1',EXIST=L1)
      IF (L1) THEN
         PRINT *,'INPUT DATA FILE FOUND!'
         PRINT *
         PRINT *,'DO YOU WISH TO USE IT FOR INPUT?'
         READ (*,*) ANS
         IF(INDEX(ANS,'Y').GT.0.OR.INDEX(ANS,'y').GT.0) THEN

C *** READ IN DATA FROM EXISTING DATA FILE
            OPEN(10,FILE='/FIRE DATA B1',STATUS='OLD')
            READ(10,*) X,Y,H,TFLR,TWAL,TA
            READ(10,*) NI,NJ,NK
            READ(10,*) NCHIP,NMS,NWRP,NTHCO
            READ(10,*) TMAX,DTIME,TTAPE,TWRITE,HSTART
            READ(10,*) NHSZ(1,1),NHSZ(1,2),NHSZ(2,1),NHSZ(2,2),
     &                 NHSZ(3,1),NHSZ(3,2)
            IF (NCHIP.LE.0) GOTO 33
            DO 32 N=1,NCHIP
               READ(10,*) ICHPB(N),NCHPI(N),JCHPB(N),NCHPJ(N),KCHPB(N),
     &                    NCHPK(N),CPS(N),CONS(N),WFAN(N)
```

```
   32       CONTINUE
   33       IF (NMS.LE.0) GOTO 37
            DO 36 N=1,NMS
               READ(10,*) IMSB(N),NMSI(N),JMSB(N),NMSJ(N),KMSB(N),
     &                    NMSK(N),RMS(N)
   36       CONTINUE
   37       DO 38 I=1,NTHCO
               READ (10,*) CX(I),CY(I),CZ(I)
   38       CONTINUE

C *** UPDATE EXISTING DATA

C *** UPDATE COMPARTMENT DIMENSIONS
            PRINT *,'CURRENT DATA:'
            PRINT *
            PRINT *,'COMPARTMENT DIMENSIONS'
            PRINT *,'     LENGTH (X DIRECTION)    : ',X,' FEET'
            PRINT *,'     WIDTH  (Y DIRECTION)    : ',Y,' FEET'
            PRINT *,'     HEIGHT (Z DIRECTION)    : ',H,' FEET'
            PRINT *,'     WALL THICKNESS          : ',TWAL,' INCHES'
            PRINT *,'     FLOOR/CEILING THICKNESS: ',TFLR,' INCHES'
            PRINT *
            PRINT *,'DO YOU WISH TO CHANGE ANY OF THESE VALUES?'
            READ(5,*) ANS
            IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
               PRINT *,'DO YOU WISH TO CHANGE THE LENGTH?'
               READ(5,*) ANS
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                  PRINT *,'ENTER NEW LENGTH:'
                  READ(5,*) X
               ENDIF
               PRINT *,'DO YOU WISH TO CHANGE THE HEIGHT?'
               READ(5,*) ANS
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                  PRINT *,'ENTER NEW HEIGHT:'
                  READ(5,*) H
               ENDIF
               PRINT *,'DO YOU WISH TO CHANGE THE WIDTH?'
               READ(5,*) ANS
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                  PRINT *,'ENTER NEW WIDTH:'
                  READ(5,*) Y
               ENDIF
               PRINT *,'DO YOU WISH TO CHANGE THE WALL THICKNESS?'
               READ(5,*) ANS
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                  PRINT *,'ENTER NEW WALL THICKNESS:'
                  READ(5,*) TWAL
               ENDIF
               PRINT *,
     &            'DO YOU WISH TO CHANGE THE FLOOR/CEILING THICKNESS?'
               READ(5,*) ANS
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                  PRINT *,'ENTER NEW FLOOR/CEILING THICKNESS:'
                  READ(5,*) TFLR
               ENDIF
```

```
            ENDIF

C *** UPDATE NUMBER OF CELLS
            PRINT *
            PRINT *,'CURRENT DATA:'
            PRINT *
            PRINT *,'NUMBER OF CELLS:'
            PRINT *,'      X DIRECTION:     ',NI
            PRINT *,'      Y DIRECTION:     ',NJ
            PRINT *,'      Z DIRECTION:     ',NK
            PRINT *
            PRINT *,'DO YOU WISH TO CHANGE ANY OF THESE VALUES?'
            READ(5,*) ANS
            IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
               PRINT *,'DO YOU WISH TO CHANGE THE NUMBER OF X CELLS?'
               READ(5,*) ANS
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                  PRINT *,'ENTER NUMBER OF X CELLS:'
                  READ(5,*) NI
               ENDIF
               PRINT *,'DO YOU WISH TO CHANGE THE NUMBER OF Y CELLS?'
               READ(5,*) ANS
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                  PRINT *,'ENTER NUMBER OF Y CELLS:'
                  READ(5,*) NJ
               ENDIF
               PRINT *,'DO YOU WISH TO CHANGE THE NUMBER OF Z CELLS?'
               READ(5,*) ANS
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                  PRINT *,'ENTER NUMBER OF Z CELLS:'
                  READ(5,*) NK
               ENDIF
            ENDIF

C *** UPDATE INTERNAL SOLIDS
            PRINT *
            PRINT *,'CURRENT DATA:'
            PRINT *
            PRINT *,'NUMBER OF INTERNAL SOLID PIECES:  ',NCHIP
            IF(NCHIP.EQ.0) GOTO 42
            PRINT *,'PIECE  STARTING NODES  NUMBER OF NODES   ',
     &              'THERMAL    SPECIFIC  FAN'
            PRINT *,' NO.     X   Y   Z       X   Y   Z    ',
     &              'CONDUCTIVITY   HEAT    SPEED'
            DO 40 N=1,NCHIP
               PRINT 2,'  ',N,ICHPB(N)-2,JCHPB(N)-2,KCHPB(N)-2,NCHPI(N),
     &                 NCHPJ(N),NCHPK(N),CONS(N),CPS(N),WFAN(N)
    2       FORMAT (2X,I3,2(4X,3(I3,1X)),4X,F9.4,2X,F8.4,2X,F5.1)
   40       CONTINUE
   42       PRINT *,'DO YOU WISH TO CHANGE THE NUMBER OF SOLID PIECES?'
            READ(5,*) ANS
            IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
               PRINT *,'ENTER NUMBER OF INTERNAL PIECES:'
               READ(5,*) NCHIP
            ENDIF
            IF(NCHIP.EQ.0) GOTO 44
```

```
            PRINT *,
     &         'DO YOU WISH TO CHANGE POSITION OF THE SOLID PIECES?'
            READ(5,*) ANS
            IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
               DO 46 N=1,NCHIP
                  PRINT *,'FOR SOLID PIECE NUMBER ',N,' ENTER THE'
                  PRINT *,'FIRST NODE IN EACH DIRECTION (X, Y AND Z)'
                  READ(5,*) ICHPB(N),JCHPB(N),KCHPB(N)
                  ICHPB(N)=ICHPB(N)+2
                  JCHPB(N)=JCHPB(N)+2
                  KCHPB(N)=KCHPB(N)+2
                  PRINT *,
     &               'NUMBER OF NODES IN EACH DIRECTION (X, Y AND Z)'
                  READ(5,*) NCHPI(N),NCHPJ(N),NCHPK(N)
                  PRINT *,'THERMAL CONDUCTIVITY'
                  READ(5,*) CONS(N)
                  PRINT *,'SPECIFIC HEAT'
                  READ(5,*) CPS(N)
                  PRINT *,'AND FAN VELOCITY (0 IF NO FAN)'
                  READ(5,*) WFAN(N)
   46          CONTINUE
            ENDIF

C *** UPDATE MASS SOURCE DATA
   44       PRINT *
            PRINT *,'CURRENT DATA:'
            PRINT *
            PRINT *,'NUMBER OF MASS SOURCES:  ',NMS
            IF(NMS.EQ.0) GOTO 50
            PRINT *,'SOURCE  STARTING NODE  NUMBER OF NODES  FLOW'
            PRINT *,'  NO.     X   Y   Z       X   Y   Z     RATE'
            DO 52 N=1,NMS
               PRINT 4,N,IMSB(N)-2,JMSB(N)-2,KMSB(N)-2,NMSI(N),
     &                 NMSJ(N),NMSK(N),
     &                 RMS(N)*(60.*H**2*U0*NMSI(N)*NMSJ(N)*NMSK(N))
     4         FORMAT (3X,I3,4X,2(3(1X,I3),4X),F5.1)
   52       CONTINUE
   50       PRINT *,'DO YOU WISH TO CHANGE THE NUMBER OF SOURCES?'
            READ(5,*) ANS
            IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
               PRINT *,'ENTER NUMBER OF SOURCES:'
               READ(5,*) NMS
            ENDIF
            IF(NMS.EQ.0) GOTO 54
            PRINT *,
     &        'DO YOU WISH TO CHANGE THE DATA FOR THE MASS SOURCES?'
            READ(5,*) ANS
            IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
               DO 56 N=1,NMS
                  PRINT *
                  PRINT *,'FOR MASS SOURCE NUMBER ',N,' ENTER'
                  PRINT *,'FIRST NODE IN EACH DIRECTION (X, Y AND Z)'
                  READ(5,*) IMSB(N),JMSB(N),KMSB(N)
                  IMSB(N)=IMSB(N)+2
                  JMSB(N)=JMSB(N)+2
                  KMSB(N)=KMSB(N)+2
```

```
                  PRINT *,
     &               'NUMBER OF NODES IN EACH DIRECTION (X, Y AND Z)'
                  READ(5,*) NMSI(N),NMSJ(N),NMSK(N)
                  PRINT *,'THE FLOW RATE OF THE MASS SOURCE IN CFM'
                  READ(5,*) RMS(N)
                  RMS(N)=RMS(N)/(60.*H**2*U0*NMSI(N)*NMSJ(N)*NMSK(N))
   56          CONTINUE
               ENDIF

C *** UPDATE THERMOCOUPLE DATA
   54          PRINT *
               PRINT *,'CURRENT DATA:'
               PRINT *
               PRINT *,'NUMBER OF THERMOCOUPLES:  ',NTHCO
               IF(NTHCO.EQ.0) GOTO 60
               PRINT *,'TC                LOCATION'
               PRINT *,'NO.        X          Y          Z'
               DO 62 N=1,NTHCO
                  PRINT ('(1X,I2,2X,3(2X,F8.4))'),
     &               N,CX(N)*H,CY(N)*H,CZ(N)*H
   62          CONTINUE
   60          PRINT *,'DO YOU WISH TO CHANGE THE NUMBER OF THERMOCOUPLES?'
               READ(5,*) ANS
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                  PRINT *,'ENTER THE NUMBER OF THERMOCOUPLES:'
                  READ(5,*) NTHCO
               ENDIF
               IF(NTHCO.EQ.0) GOTO 70
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) GOTO 65
               PRINT *,'DO YOU WISH TO CHANGE THE THERMOCOUPLE LOCATIONS?'
               READ(5,*) ANS
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) GOTO 65
               GOTO 70
   65          DO 66 N=1,NTHCO
                  PRINT *,'ENTER THE LOCATION (FEET) OF THERMOCOUPLE ',N
                  READ(5,*) XCX,XCY,XCZ
                  CX(N)=XCX/H
                  CY(N)=XCY/H
                  CZ(N)=XCZ/H
   66          CONTINUE

C *** UPDATE HEAT SOURCE DATA
   70          PRINT *
               PRINT *,'CURRENT DATA'
               PRINT *
               PRINT *,'HEAT SOURCE LOCATION'
               PRINT *,' X DIRECTION:  NODE',NHSZ(1,1)-2,' TO',NHSZ(1,2)-2
               PRINT *,' Y DIRECTION:  NODE',NHSZ(2,1)-2,' TO',NHSZ(2,2)-2
               PRINT *,' Z DIRECTION:  NODE',NHSZ(3,1)-2,' TO',NHSZ(3,2)-2
               PRINT *
               PRINT *,'DO YOU WISH TO CHANGE THE STARTING COORDINATES?'
               READ(5,*) ANS
               IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                  PRINT *,'ENTER THE STARTING COORDINATES:'
                  PRINT *,'   X DIRECTION:'
                  READ(5,*) NHSZ(1,1)
```

```
                PRINT *,'   Y DIRECTION:'
                READ(5,*) NHSZ(2,1)
                PRINT *,'   Z DIRECTION:'
                READ(5,*) NHSZ(3,1)
                DO 71 I=1,3
                   NHSZ(I,1)=NHSZ(I,1)+2
   71           CONTINUE
             ENDIF
             PRINT *,'DO YOU WISH TO CHANGE THE ENDING COORDINATES?'
             READ(5,*) ANS
             IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                PRINT *,'ENTER THE ENDING COORDINATES:'
                PRINT *,'   X DIRECTION:'
                READ(5,*) NHSZ(1,2)
                PRINT *,'   Y DIRECTION:'
                READ(5,*) NHSZ(2,2)
                PRINT *,'   Z DIRECTION:'
                READ(5,*) NHSZ(3,2)
                DO 72 I=1,3
                   NHSZ(I,2)=NHSZ(I,2)+2
   72           CONTINUE
             ENDIF

C *** UPDATE AMBIENT TEMPERATURE
             PRINT *
             PRINT *,'CURRENT DATA'
             PRINT *
             PRINT 3,
     &          'AMBIENT TEMPERATURE =',TA-459.67,'DEGREES FARENHEIT'
    3        FORMAT (1X,A,1X,F6.2,1X,A)
             PRINT *
             PRINT *,'DO YOU WISH TO CHANGE THE AMBIENT TEMPERATURE?'
             READ(5,*) ANS
             IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                PRINT *,'ENTER THE NEW AMBIENT TEMPERATURE IN DEGREES F:'
                READ(5,*) TAF
                TA=TAF+459.67
             ENDIF

C *** UPDATE TIME DATA
             PRINT *
             PRINT *,'CURRENT DATA'
             PRINT *
             PRINT *,'MAXIMUM RUN TIME         =',TMAX*H/U0,' SECONDS'
             PRINT *,'INCREMENTAL TIME STEP    =',DTIME*H/U0,' SECONDS'
             PRINT *,'TIME BETWEEN DATA OUTPUT =',TWRITE,' SECONDS'
             PRINT *,'TIME BETWEEN PLOTS       =',TTAPE,' SECONDS'
             PRINT *,'FIRE START TIME          =',HSTART,' SECONDS'
             PRINT *
             PRINT *,'DO YOU WISH TO CHANGE ANY OF THESE TIMES?'
             READ(5,*) ANS
             IF(INDEX(ANS,'Y').NE.0.OR.INDEX(ANS,'y').NE.0) THEN
                PRINT *,'ENTER MAXIMUM RUN TIME IN SECONDS:'
                READ(5,*) XTMAX
                TMAX=XTMAX*U0/H
                PRINT *,'INCREMENTAL TIME STEP IN SECONDS:'
```

```
                READ(5,*) XDTIME
                DTIME=XDTIME*U0/H
                PRINT *,'TIME BETWEEN DATA OUTPUT IN SECONDS:'
                READ(5,*) TWRITE
                PRINT *,'TIME BETWEEN PLOTS IN SECONDS:'
                READ(5,*) TTAPE
                PRINT *,'FIRE START TIME IN SECONDS:'
                READ(5,*) HSTART
                NWRP=INT(TWRITE/XDTIME)/2
             ENDIF
          ELSE
              PRINT *,'PROGRAM TERMINATING.'
              PRINT *,'PLEASE RENAME FIRE.DAT AND TRY AGAIN.'
              GOTO 999
          ENDIF

***** CREATING NEW DATA FILE

       ELSE
          PRINT *,'FIRE DATA B1 NOT FOUND!'
          PRINT *,'CREATING NEW DATA FILE.'
          OPEN(10,FILE='/FIRE DATA B1',STATUS='NEW')

***** INPUT NEW DATA *****

C *** INPUT GEOMETRIC DATA
          PRINT *,'INPUT COMPARTMENT LENGTH (X DIRECTION) IN FEET'
          READ(5,*) X
          PRINT *,'INPUT COMPARTMENT HEIGHT (Z DIRECTION) IN FEET'
          READ(5,*) H
          PRINT *,'INPUT COMPARTMENT WIDTH (Y DIRECTION) IN FEET'
          READ(5,*) Y
          PRINT *,'INPUT COMPARTMENT WALL THICKNESS IN INCHES'
          READ(5,*) TWAL
          PRINT *,'INPUT COMPARTMENT FLOOR/CEILING THICKNESS IN INCHES'
          READ(5,*) TFLR
          PRINT *,'INPUT NUMBER OF CELLS IN THE X DIRECTION:'
          READ(5,*) NI
          PRINT *,'INPUT NUMBER OF CELLS IN THE Y DIRECTION:'
          READ(5,*) NJ
          PRINT *,'INPUT NUMBER OF CELLS IN THE Z DIRECTION:'
          READ(5,*) NK

C *** INPUT INTERNAL SOLIDS DATA
          PRINT *,'INPUT NUMBER OF INTERNAL SOLID PIECES'
          READ(5,*) NCHIP
          IF (NCHIP.GT.0) THEN
             PRINT *,'INPUT THE LOCATION OF INTERNAL SOLID PIECES'
             DO 14 N=1,NCHIP
                PRINT *
                PRINT *,'FOR SOLID PIECE NUMBER ',N,' ENTER THE'
                PRINT *,'FIRST NODE IN EACH DIRECTION (X, Y AND Z)'
                READ(5,*) ICHPB(N),JCHPB(N),KCHPB(N)
                ICHPB(N)=ICHPB(N)+2
                JCHPB(N)=JCHPB(N)+2
                KCHPB(N)=KCHPB(N)+2
```

```
                PRINT *,'NUMBER OF NODES IN EACH DIRECTION (X, Y AND Z)'
                READ(5,*) NCHPI(N),NCHPJ(N),NCHPK(N)
                PRINT *,'THERMAL CONDUCTIVITY'
                READ(5,*) CONS(N)
                PRINT *,'SPECIFIC HEAT'
                READ(5,*) CPS(N)
                PRINT *,'AND FAN VELOCITY (0 IF NO FAN)'
                READ(5,*) WFAN(N)
   14        CONTINUE
             ENDIF

C *** INPUT MASS SOURCE DATA
             PRINT *,'INPUT NUMBER OF MASS SOURCES'
             READ(5,*) NMS
             IF (NMS.EQ.0) THEN
                PRINT *
                PRINT *,'ENTER MASS SOURCE DATA:'
                DO 18 N=1,NMS
                   PRINT *,'FIRST NODE IN EACH DIRECTION (X, Y AND Z)'
                   READ(5,*) IMSB(N),JMSB(N),KMSB(N)
                   IMSB(N)=IMSB(N)+2
                   JMSB(N)=JMSB(N)+2
                   KMSB(N)=KMSB(N)+2
                   PRINT *,'NUMBER OF NODES IN EACH DIRECTION (X, Y AND Z)'
                   READ(5,*) NMSI(N),NMSJ(N),NMSK(N)
                   PRINT *,'THE FLOW RATE OF THE MASS SOURCE IN CFM'
                   READ(5,*) RMS(N)
                   RMS(N)=RMS(N)/(60.*H**2*U0*NMSI(N)*NMSJ(N)*NMSK(N))
   18        CONTINUE
             ENDIF

C *** INPUT THERMOCOUPLE DATA
             PRINT *,'INPUT NUMBER OF THERMOCOUPLES'
             READ(5,*) NTHCO
             PRINT *,'INPUT THERMOCOUPLE POSITION IN FEET:'
             DO 19 I=1,NTHCO
                PRINT *
                PRINT *,'FOR THERMOCOUPLE NUMBER ',I
                PRINT *,'      X POSITION (FEET):'
                READ (5,*) XCX
                CX(I)=XCX/H
                PRINT *,'      Y POSITION (FEET):'
                READ (5,*) XCY
                CY(I)=XCY/H
                PRINT *,'      Z POSITION (FEET):'
                READ (5,*) XCZ
                CZ(I)=XCZ/H
   19        CONTINUE

C *** INPUT HEAT SOURCE DATA
             PRINT *,'INPUT FIRST NODE OF HEAT SOURCE '
             PRINT *,'      X DIRECTION:'
             READ (5,*) NHSZ(1,1)
             PRINT *,'      Y DIRECTION:'
             READ (5,*) NHSZ(2,1)
             PRINT *,'      Z DIRECTION:'
```

```
         READ (5,*) NHSZ(3,1)
         PRINT *,'INPUT LAST NODE OF HEAT SOURCE IN EACH DIRECTION:'
         PRINT *,'      X DIRECTION:'
         READ (5,*) NHSZ(1,2)
         PRINT *,'      Y DIRECTION:'
         READ (5,*) NHSZ(2,2)
         PRINT *,'      Z DIRECTION:'
         READ (5,*) NHSZ(3,2)

C *** CORRECT HEAT SOURCE LOCATION DUE TO EXTERNAL CELLS
         DO 5 I=1,3
            DO 6 J=1,2
               NHSZ(I,J)=NHSZ(I,J)+2
    6       CONTINUE
    5    CONTINUE

C *** INPUT AMBIENT TEMPERATURE
         PRINT *,'INPUT AMBIENT TEMPERATURE (DEGREES FARENHEIT)'
         READ(5,*) TAF
         TA=TAF+459.67

C *** INPUT TIME DATA
         PRINT *,'INPUT MAX RUN TIME FOR FIRE (SECONDS)'
         READ(5,*) XTMAX
         TMAX=XTMAX*U0/H
         PRINT *,'INPUT SIZE OF TIME STEP (SECONDS)'
         READ(5,*) XDTIME
         DTIME=XDTIME*U0/H
         PRINT *,'INPUT FIRE START TIME (SECONDS)'
         READ(5,*) HSTART
         PRINT *,'INPUT TIME INTERVAL BETWEEN DATA SAVES (SECONDS)'
         READ(5,*) TWRITE
         PRINT *,'INPUT TIME INTERVAL BETWEEN PLOTS (SECONDS)'
         READ(5,*) TTAPE

C *** DETERMINE NWRP AND ADJUST TWRITE
         NWRP=INT(TWRITE/XDTIME)/2
      ENDIF

C *** SAVE DATA IN DATA FILE
      REWIND(10)
      WRITE(10,*) X,Y,H,TFLR,TWAL,TA
      WRITE(10,*) NI,NJ,NK
      WRITE(10,*) NCHIP,NMS,NWRP,NTHCO
      WRITE(10,*) TMAX,DTIME,TTAPE,TWRITE,HSTART
      WRITE(10,*) NHSZ(1,1),NHSZ(1,2),NHSZ(2,1),NHSZ(2,2),NHSZ(3,1),
     &            NHSZ(3,2)
      IF (NCHIP.EQ.0) GOTO 20
      DO 22 N=1,NCHIP
         WRITE(10,*) ICHPB(N),NCHPI(N),JCHPB(N),NCHPJ(N),KCHPB(N),
     &               NCHPK(N),CPS(N),CONS(N),WFAN(N)
   22 CONTINUE
   20 IF (NMS.EQ.0) GOTO 24
      DO 26 N=1,NMS
         WRITE(10,*) IMSB(N),NMSI(N),JMSB(N),NMSJ(N),KMSB(N),
     &               NMSK(N),RMS(N)
```

```
 26 CONTINUE
 24 DO 28 I=1,NTHCO
       WRITE(10,*) CX(I),CY(I),CZ(I)
 28 CONTINUE
    REWIND(10)
    CLOSE(10)

999 END
```

APPENDIX B. PROGRAM FIRE

```
      PROGRAM FIRE
C*************************************************************************
C**                                                                     **
C**               THREE-DIMENSIONAL NUMERICAL SIMULATION                **
C**                 OF A FIRE SPREAD INSIDE A BUILDING                  **
C**                                                                     **
C**                          DEVELOPED BY :                             **
C**                      H.Q. YANG AND K.T. YANG                        **
C**                                                                     **
C**         DEPARTMENT OF AEROSPACE & MECHANICAL ENGINEERING            **
C**                      UNIVERSITY OF NOTRE DAME                       **
C**                     NOTRE DAME, INDIANA, 46556                      **
C**                                                                     **
C**                            DEC. 1986                                **
C**                                                                     **
C**                           AMMENDED BY                               **
C**                        K.J. THORKILDSEN                             **
C**                    LIEUTENANT, U.S. COAST GUARD                     **
C**                                                                     **
C**                  DEPARTMENT OF MECHANICAL ENGINEERING               **
C**                       NAVAL POSTGRADUATE SCHOOL                     **
C**                      MONTEREY, CALIFORNIA  93942                    **
C**                                                                     **
C**                            SEP. 1992                                **
C**                                                                     **
C*************************************************************************
C*SET CONSTANTS:
C* CP0     : REFERENCE SPECIFIC HEAT OF AIR =
C* GC      : GRAVITATIONAL ACCELERATION = 32.17 FT/SEC**2
C* RAIR    : UNIVERSAL GAS CONSTANT FOR AIR = 53.34
C* RHO0    : REFERENCE AIR DENSITY (LBM/FT**3) = 0.0714 LBM/FT**3
C* U0      : REFERENCE VELOCITY (FT/SEC) = 1.0 FT/SEC
C*
C*COMPARTMENT DIMENSIONS (IN FEET):
C* H       : HEIGHT IN Z-DIRECTION (USED AS REFERENCE LENGTH)
C* X       : LENGTH IN X-DIRECTION
C* Y       : WIDTH IN Y-DIRECTION
C*
C* NI      : NUMBER OF CELLS IN X-DIRECTION
C* NJ      :                    Y-DIRECTION
C* NK      :                    Z-DIRECTION
C*
C* CONSRA  : TA**3/(RA*CP*U0*H*H)
C* HCONV   : HEAT TRANSFER COEFFICIENT TO THE AMBIENT (BTU/H*K*FT**2)
C* TA      : REFERENCE TEMPERATURE (R)
C* TINIT   : INITIAL TEMPERATURE (0)
C* UR      : REFERENCE VELOCITY (CM/S)
C*
C*HEAT SOURCE DATA:
C* NHSZ(1,1) : STARTING CONTROL VOLUME NUMBER IN X-DIRECTION
C* NHSZ(2,1) :                                  Y-DIRECTION
```

```
* NHSZ(3,1) :                                        Z-DIRECTION
* NHSZ(1,2) : LAST CONTROL VOLUME NUMBER IN X-DIRECTION
* NHSZ(2,2) :                                      Y-DIRECTION
* NHSZ(3,2) :                                      Z-DIRECTION
*
*INTERNAL SOLID PIECES:
* NCHIP   : NUMBER OF INTERNAL SOLID PIECES
* ICHPB() : STARTING NODE NUMBER FOR SOLID IN X-DIRECTION
* JCHPB() :                                    Y-DIRECTION
* KCHPB() :                                    Z-DIRECTION
* NCHPI() : NUMBER OF NODES OF SOLID IN X-DIRECTION
* NCHPJ() :                                 Y-DIRECTION
* NCHPK() :                                 Z-DIRECTION
*
*TOTAL HEAT:
* QSIN    : INPUT FROM THE FIRE
* QSWAL   : LOST TO THE WALL (FROM AIR TO THE WALL)
* QSFAN   : CARRIED AWAY BY THE VENTILATION
*
*VIEW FACTORS FROM HEAT SOURCE:
* VFHSW(N,J,K) : TO ELEMENT J,K ON WEST  WALL
* VFHSE(N,J,K) :                EAST  WALL
* VFHSN(N,K,I) : TO ELEMENT K,I ON NORTH WALL
* VFHSS(N,K,I) :                SOUTH WALL
* VFHSF(N,I,J) : TO ELEMENT I,J ON FRONT WALL
* VFHSB(N,I,J) :                BACK  WALL
*******************************************************************************
*DATA FILES USED IN THIS PROGRAM:
*
* FILE # 10 = FIRE DATA     : INITIAL SET-UP DATA
*        11 = CONTINUE DATA : RESTART/CONTINUATION DATA
*        12 = OUTPUT DATA   : OUTPUT RESULTS
*        13 = PLOT DATA     : DATA FOR PLOTTING
*******************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(25),YC(25),ZC(25),XS(25),YS(25),ZS(25),DXXC(25),
     &          DYYC(25),DZZC(25),DXXS(25),DYYS(25),DZZS(25)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL2/X,Y,H,TFLR,TWAL
      COMMON/BL3/F,FR,HSTART
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL12/NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,CNT,ABTURB,BTURB,VISL,VISMAX
      COMMON/BL16/U0,UGRT,BUOY,CP0,PRT,COND0,VIS0,RHO0,
     &            TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
      COMMON/BL20/SIG11(25,25,15),SIG12(25,25,15),SIG22(25,25,15),
     &            SIG13(25,25,15),SIG23(25,25,15),SIG33(25,25,15)
      COMMON/BL23/RMS(20),NMS,IMSB(20),NMSI(20),JMSB(20),NMSJ(20),
     &            KMSB(20),NMSK(20)
      COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
     &            JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
      COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
     &            COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
     &            WOD(25,25,15)
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
```

```
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
     &            CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
     &            WPD(25,25,15)
      COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
     &            SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
     &            DV(25,25,15),DW(25,25,15)
      COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
     &            AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
     &            SU(25,25,15),RI(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
      COMMON/BL38/TCOUP(30),CX(30),CY(30),CZ(30),NTH(30,3),NTHCO
      COMMON/BL39/ALEW,CONSRA,QSIN,QSWER,QSWAL,QSAIR,QSFAN
      COMMON/BL40/VFHSW(5,25,15),VFHSE(5,25,15),VFHSS(5,15,25),
     &            VFHSN(5,15,25),VFHSB(5,25,25),VFHSF(5,25,25)
      COMMON/BL41/VFHSBW(5,8,34,34),VFHSBE(5,8,34,34),VFHSBS(5,8,34,34),
     &            VFHSBN(5,8,34,34),VFHSBB(5,8,34,34),VFHSBF(5,8,34,34)
      COMMON/BL43/QSCONF,QSCONB,QSCONE,QSCONW,QSCONN,QSCONS,QSRADF,
     &            QSRADB,QSRADE,QSRADW,QSRADN,QSRADS,WAIR,WWAL,WINS,
     &            WERR,WWFAN
      DATA SORMAX,XTIME,ITMAX/3.00,0.0,4/

C ***** INITIAL PROGRAM START *****
      CALL CPUTIME(BEGIN,IPR)

C *** INPUT DATA
      CALL INPUT(NSTOP)
      IF(NSTOP.GT.0) GOTO 9999

C *** GENERATE GRID SYSTEM
      CALL GRID

C *** INITIALIZE THE ALL DATA FIELDS
      CALL INIT

C *** OPEN OUTPUT FILES
      OPEN(12,FILE='/OUTPUT DATA B1',STATUS='UNKNOWN')
      OPEN(13,FILE='/PLOT DATA B4',STATUS='UNKNOWN',
     &        FORM='UNFORMATTED')

C *** CALCULATE THE VIEW FACTORS FROM THE FIRE TO THE WALLS
      CALL VIEW

***** START CALCULATIONS *****

      NT=0
      NTIM=0
  300 NT=NT+1

C *** ON RESTART NTMAX0 IS SET EQUAL TO OLD VALUE FOR NTREAL
      IF(TIME.GE.TMAX) GO TO 277
      NTREAL=NT+NTMAX0
      TIME=TIME+DTIME
```

```
      XTIME=TIME*H/U0
C     PRINT 3,'CURRENT FIRE TIME IS:',XTIME,'SECONDS'
C   3 FORMAT (1X,A,1X,F10.6,1X,A)

C *** CALCULATE THE HEAT SOURCE IN BTU/SEC
      CALL CALQ

C *** START CALCULATIONS
      ITER=0
      JTERM=0
      JJTERM=0

C *** PREDICT VARIABLE FIELDS FOR USE BY CALVIS AND SU(I,J,K)
      DO 48 K=1,NK+4
      DO 48 J=1,NJ+4
      DO 48 I=1,NI+4
         TPD(I,J,K)=T(I,J,K)
         CPD(I,J,K)=C(I,J,K)
         RPD(I,J,K)=R(I,J,K)
         UPD(I,J,K)=U(I,J,K)
         VPD(I,J,K)=V(I,J,K)
         WPD(I,J,K)=W(I,J,K)
   48 CONTINUE
   47 JTERM=JTERM+1
  301 NTITER=0
  312 NTITER=NTITER+1

C *** IF FIRE HAS STARTED, CALCULATE THE TEMPERATURE
      IF (XTIME.GE.HSTART) CALL CALT

*****THIS STEP CAN BE SKIPPED WHEN COMPARTMENT IS OPEN TO OUTSIDE*****
C *** CORRECT GLOBAL PRESSURE FOR TOTAL MASS CONSERVATION
      CALL GLOBE

C *** CALCULATE DENSITY
      DO 100 J=1,NJ+4
      DO 100 I=1,NI+4
      DO 100 K=1,NK+4
         IF (NOD(I,J,K).EQ.1) GOTO 100
         AAAA=BUOY*UGRT*HEIGHT(I,J,K)
         R(I,J,K)=(UGRT*P(I,J,K)+(1./EXP(AAAA)))/T(I,J,K)
  100 CONTINUE

*****THIS STEP CAN BE SKIPPED WHEN COMPARTMENT IS OPEN TO OUTSIDE*****
C *** ITERATE INSIDE TEMPERATURE LOOP TO ASSURE GLOBAL CONSERVATION
C *** OF MASS AND ENERGY
      IF (NTITER.LT.2) GOTO 312

C *** PRINT OUT THE ENERGY DISTRIBUTION
*      IF (MOD(NTREAL,NWRP).EQ.0) CALL OUT(4)

C *** CALCULATE THE SMOKE CONCENTRATION
C     CALL CALC

C *** CALCULATE TURBULENT VISCOSITY AND CONDUCTIVITY
      CALL CALVIS
```

```
C *** CORRECT CONDUCTIVITY OF THE SOLID
      IF (NCHIP.NE.0) CALL SOLCON

C *** START PRESSURE CORRECTION ITERATIVE LOOP
C *** IT IS THE MAJOR PART OF THE ERROR CONTROL ROUTINE
      ITER=ITER+1

C *** CALCULATE THE STRESS AND VELOCITY COMPONENTS U,V,AND W
      CALL STRESS
      CALL CALU
      CALL CALV
      CALL CALW

C *** CALCULATE PRESSURE
      CALL CALP

C *** IF SOURCE TERM IS LARGER THAN 10.0, STOP PROGRAM
      IF (RESORM(ITER).GT.10.0) GOTO 2020

      IF(RESORM(ITER).LE.SORMAX) GO TO 49
      IF(ITER.EQ.1) GO TO 302
      IF(RESORM(ITER) .LE. RESORM(ITER-1)) GO TO 302
      GO TO 304
  302 IF(JTERM .LT. 2) THEN
         SOURCE=RESORM(ITER)
      ELSEIF(RESORM(ITER).LE.SOURCE) THEN
         SOURCE=RESORM(ITER)
      ELSE
         GOTO 304
      ENDIF
      DO 23 K=1,NK+4
      DO 23 J=1,NJ+4
      DO 23 I=1,NI+4
         TPD(I,J,K)=T(I,J,K)
         CPD(I,J,K)=C(I,J,K)
         RPD(I,J,K)=R(I,J,K)
         UPD(I,J,K)=U(I,J,K)
         VPD(I,J,K)=V(I,J,K)
         WPD(I,J,K)=W(I,J,K)
         PPD(I,J,K)=P(I,J,K)
   23 CONTINUE

      JJTERM=0
      IF(ITER.EQ.ITMAX) GO TO 49
      IF(JTERM.EQ.2) GO TO 35
      IF(ITER.EQ.4) GO TO 47
   35 IF(JTERM.EQ.3) GO TO 58
      IF(ITER.EQ.7) GO TO 47
   58 JJTERM=0
      GO TO 301
  304 JJTERM=JJTERM+1
      IF(JTERM.EQ.1) GOTO 41
      IF(JTERM.EQ.2.AND.JJTERM.EQ.1.AND.ITER.NE.5) GO TO 41
      GO TO 82
```

```
   41 DO 40 K=1,NK+4
      DO 40 J=1,NJ+4
      DO 40 I=1,NI+4
         R(I,J,K)=RPD(I,J,K)
         U(I,J,K)=UPD(I,J,K)
         V(I,J,K)=VPD(I,J,K)
         W(I,J,K)=WPD(I,J,K)
         P(I,J,K)=PPD(I,J,K)
   40 CONTINUE
      IF(ITER.EQ.ITMAX) GO TO 49
      GO TO 47

   82 DO 43 K=1,NK+4
      DO 43 J=1,NJ+4
      DO 43 I=1,NI+4
         T(I,J,K)=TPD(I,J,K)
         C(I,J,K)=CPD(I,J,K)
         R(I,J,K)=RPD(I,J,K)
         U(I,J,K)=UPD(I,J,K)
         V(I,J,K)=VPD(I,J,K)
         W(I,J,K)=WPD(I,J,K)
         P(I,J,K)=PPD(I,J,K)
   43 CONTINUE

      IF(ITER.EQ.ITMAX) GO TO 49
      IF((JTERM.EQ.3.AND.ITER.NE.8).OR.JJTERM.EQ.2) GO TO 49
      GO TO 301

   49 ITERT=ITERT+ITER
*       IF (MOD(NTREAL,NWRP).EQ.0) CALL OUT(1)

C *** FIND TEMPERATURES AT THERMOCOUPLES AND PRINT OUT AT PROPER TIME
      CALL TCP
      IF (MOD(NTREAL,NWRITE).EQ.0) CALL OUT(2)

C *** OUTPUT FILED VALUES
      IF (MOD(NTREAL,NWRITE).EQ.0) CALL OUT(3)
      IF(TIME.GE.TMAX) GO TO 277

C *** SHIFT CURRENT TIME VALUES TO PREVIOUS TIME VALUES AND
C *** LOOP BACK FOR NEXT ITERATION
      DO 305 K=1,NK+4
      DO 305 J=1,NJ+4
      DO 305 I=1,NI+4
         TOD(I,J,K)=T(I,J,K)
         COD(I,J,K)=C(I,J,K)
         ROD(I,J,K)=R(I,J,K)
         UOD(I,J,K)=U(I,J,K)
         VOD(I,J,K)=V(I,J,K)
         WOD(I,J,K)=W(I,J,K)
         POD(I,J,K)=P(I,J,K)
  305 CONTINUE

C *** OUTPUT TO DATA FILE FOR PLOTTING
      IF(MOD(NTREAL,NTAPE).EQ.0) THEN
         WRITE(13) TIME,T,U,V,W
```

```
      ENDIF

C *** OUTPUT TO CONTINUATION FILE FOR RESTART
      IF(MOD(NTREAL,100).EQ.0) THEN
         WRITE(11) TIME,NTREAL,FR,T,R,U,V,W,P,C
         REWIND 11
      ENDIF
      GO TO 300

C *** OUTPUT TO CONTINUATION FILE
  277 WRITE(11) TIME,NTREAL,FR,T,R,U,V,W,P,C
      GO TO 9999

 2020 WRITE(12,*)'RESIDUAL MASS IS LARGER THAN 10.0',
     &            ' PROGRAM STOPS AT TIME = ',XTIME,' SEC'

 9999 CALL CPUTIME(END,IPR)
      WRITE (12,*) 'CPU RUN TIME = ',(END-BEGIN)*1.E-6,' SECONDS'
      STOP
      END

*******************************************************************************
*******************************************************************************
      BLOCK DATA
*******************************************************************************
* U0      : REFERENCE VELOCITY = 1.0 FT/SEC
* PRT     : TURBULENT PRANDTL NUMBER = 1.0
* RHO0    : REFERENCE DENSITY OF AIR = 0.0714 LBM/FT**3
* CP0     : REFERENCE SPECIFIC HEAT OF AIR = 0.24 BTU/(LBM*F)
* VIS0    : REFERENCE VISCOSITY = 1.56E-4
* CNT     :
* ABTURB  : TURBULENCE CONSTANT
* BTURB   : TURBULENCE CONSTANT
* GC      : GRAVITATIONAL ACCELERATION = 32.17 FT/SEC**2
* RAIR    : GAS CONSTANT FOR AIR = 53.34
* ALEW    : LEWIS NUMBER = 1.0
*******************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/BL12/NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,CNT,ABTURB,BTURB,VISL,VISMAX
      COMMON/BL16/U0,UGRT,BUOY,CP0,PRT,COND0,VIS0,RHO0,
     &            TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
      COMMON/BL39/ALEW,CONSRA,QSIN,QSWER,QSWAL,QSAIR,QSFAN

C *** SPECIFY THE INITIAL DATA
      DATA  U0, PRT,  RHO0 ,  CP0,   VIS0  , NTMAX0/
     &     1.0, 1.0, 0.0714, 0.24,  1.56D-4,   0/
      DATA CNT,ABTURB,BTURB/0.2,2.0,1.0/
      DATA GC,RAIR,ALEW/32.17,53.34,1.0/
      END

*******************************************************************************
*******************************************************************************
```

```
      SUBROUTINE CALC
*******************************************************************************
*THIS SUBROUTINE CALCULATES THE SMOKE CONCENTRATIONS

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL2/X,Y,H,TFLR,TWAL
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
     &            COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
     &            WOD(25,25,15)
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
     &            CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
     &            WPD(25,25,15)
      COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
     &            AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
     &            SU(25,25,15),RI(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
      COMMON/BL39/ALEW,CONSRA,QSIN,QSWER,QSWAL,QSAIR,QSFAN

C *** CALCULATE COEFFICIENTS
      DO 100 K=2,NK+3
      DO 100 J=2,NJ+3
      DO 100 I=2,NI+3

C *** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
         DXP1=DXXC(I+1)
         DXI =DXXC(I)
         DXM1=DXXC(I-1)

         DYP1=DYYC(J+1)
         DYJ =DYYC(J)
         DYM1=DYYC(J-1)

         DZP1=DZZC(K+1)
         DZK =DZZC(K)
         DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
         DXN=DXXC(I)
         DXS=DXXC(I)
         DXF=DXXC(I)
         DXB=DXXC(I)

         DYF=DYYC(J)
         DYB=DYYC(J)
         DYE=DYYC(J)
         DYW=DYYC(J)

         DZE=DZZC(K)
         DZW=DZZC(K)
```

```
          DZN=DZZC(K)
          DZS=DZZC(K)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T
          DXEE=DXXS(I+2)
          DXE =DXXS(I+1)
          DXW =DXXS(I)
          DXWW=DXXS(I-1)

          DYNN=DYYS(J+2)
          DYN =DYYS(J+1)
          DYS =DYYS(J)
          DYSS=DYYS(J-1)

          DZFF=DZZS(K+2)
          DZF =DZZS(K+1)
          DZB =DZZS(K)
          DZBB=DZZS(K-1)

C *** DEFINE THE AREA OF THE CONTROL VOLUME
          DXYF=DXF*DYF
          DXYB=DXB*DYB
          DYZE=DYE*DZE
          DYZW=DYW*DZW
          DZXN=DZN*DXN
          DZXS=DZS*DXS

          VOL=DXI*DYJ*DZK
          VOLDT=VOL/DTIME

          ZXOYN=DZXN/DYN
          ZXOYS=DZXS/DYS
          XYOZF=DXYF/DZF
          XYOZB=DXYB/DZB
          YZOXE=DYZE/DXE
          YZOXW=DYZW/DXW

C *** DENSITY AT THE SURFACES OF THE CONTROL VOLUME
          GN=(R(I,J,K)*DYP1+R(I  ,J+1,K  )*DYJ)/(DYP1+DYJ)
          GS=(R(I,J,K)*DYM1+R(I  ,J-1,K  )*DYJ)/(DYM1+DYJ)
          GE=(R(I,J,K)*DXP1+R(I+1,J  ,K  )*DXI)/(DXP1+DXI)
          GW=(R(I,J,K)*DXM1+R(I-1,J  ,K  )*DXI)/(DXM1+DXI)
          GF=(R(I,J,K)*DZP1+R(I  ,J  ,K+1)*DZK)/(DZP1+DZK)
          GB=(R(I,J,K)*DZM1+R(I  ,J  ,K-1)*DZK)/(DZM1+DZK)

          CN=GN*V(I  ,J+1,K  )*DZXN
          CS=GS*V(I  ,J  ,K  )*DZXS
          CE=GE*U(I+1,J  ,K  )*DYZE
          CW=GW*U(I  ,J  ,K  )*DYZW
          CF=GF*W(I  ,J  ,K+1)*DXYF
          CB=GB*W(I  ,J  ,K  )*DXYB

C *** DIFFUSIVITY AT THE SURFACES OF THE CONTROL VOLUME
          CONDN=(DYP1+DYJ)/(DYJ/COND(I,J,K)+DYP1/COND(I  ,J+1,K  ))
          CONDS=(DYM1+DYJ)/(DYJ/COND(I,J,K)+DYM1/COND(I  ,J-1,K  ))
          CONDE=(DXP1+DXI)/(DXI/COND(I,J,K)+DXP1/COND(I+1,J  ,K  ))
```

```
      CONDW=(DXM1+DXI)/(DXI/COND(I,J,K)+DXM1/COND(I-1,J  ,K  ))
      CONDF=(DZP1+DZK)/(DZK/COND(I,J,K)+DZP1/COND(I  ,J  ,K+1))
      CONDB=(DZM1+DZK)/(DZK/COND(I,J,K)+DZM1/COND(I  ,J  ,K-1))

      CONDN1=ZXOYN*CONDN*ALEW
      CONDS1=ZXOYS*CONDS*ALEW
      CONDE1=YZOXE*CONDE*ALEW
      CONDW1=YZOXW*CONDW*ALEW
      CONDF1=XYOZF*CONDF*ALEW
      CONDB1=XYOZB*CONDB*ALEW

C *** QUICK SCHEME
      CEP=(ABS(CE)+CE)*DXP1*DXI/(DXE*(DXE+DXW ))/8.
      CEM=(ABS(CE)-CE)*DXP1*DXI/(DXE*(DXE+DXEE))/8.
      CWP=(ABS(CW)+CW)*DXM1*DXI/(DXW*(DXW+DXWW))/8.
      CWM=(ABS(CW)-CW)*DXM1*DXI/(DXW*(DXW+DXE ))/8.

      CNP=(ABS(CN)+CN)*DYP1*DYJ/(DYN*(DYN+DYS ))/8.
      CNM=(ABS(CN)-CN)*DYP1*DYJ/(DYN*(DYN+DYNN))/8.
      CSP=(ABS(CS)+CS)*DYM1*DYJ/(DYS*(DYS+DYSS))/8.
      CSM=(ABS(CS)-CS)*DYM1*DYJ/(DYS*(DYS+DYN ))/8.

      CFP=(ABS(CF)+CF)*DZP1*DZK/(DZF*(DZF+DZB ))/8.
      CFM=(ABS(CF)-CF)*DZP1*DZK/(DZF*(DZF+DZFF))/8.
      CBP=(ABS(CB)+CB)*DZM1*DZK/(DZB*(DZB+DZBB))/8.
      CBM=(ABS(CB)-CB)*DZM1*DZK/(DZB*(DZB+DZF ))/8.

      AE(I,J,K)=-.5*CE*DXI/DXE+CEP+CEM*(1.+DXE/DXEE)+CWM*DXW/DXE
      AW(I,J,K)= .5*CW*DXI/DXW+CWM+CWP*(1.+DXW/DXWW)+CEP*DXE/DXW
      AN(I,J,K)=-.5*CN*DYJ/DYN+CNP+CNM*(1.+DYN/DYNN)+CSM*DYS/DYN
      AS(I,J,K)= .5*CS*DYJ/DYS+CSM+CSP*(1.+DYS/DYSS)+CNP*DYN/DYS
      AF(I,J,K)=-.5*CF*DZK/DZF+CFP+CFM*(1.+DZF/DZFF)+CBM*DZB/DZF
      AB(I,J,K)= .5*CB*DZK/DZB+CBM+CBP*(1.+DZB/DZBB)+CFP*DZF/DZB

C *** BOUNDARY CONSIDERATION
      IF (I.LT.NI+3) THEN
         AEE=-CEM*DXE/DXEE
         AEER=AEE*CPD(I+2,J,K)
      ELSE
         AEE=0.
         AEER=0.
      ENDIF

      IF (I.GT.2) THEN
         AWW=-CWP*DXW/DXWW
         AWWR=AWW*CPD(I-2,J,K)
      ELSE
         AWW=0.
         AWWR=0.
      ENDIF

      IF (J.LT.NJ+3) THEN
         ANN=-CNM*DYN/DYNN
         ANNR=ANN*CPD(I,J+2,K)
      ELSE
         ANN=0.
```

```
            ANNR=0.
         ENDIF

         IF (J.GT.2) THEN
            ASS=-CSP*DYS/DYSS
            ASSR=ASS*CPD(I,J-2,K)
         ELSE
            ASS=0.
            ASSR=0.
         ENDIF

         IF (K.LT.NK+3) THEN
            AFF=-CFM*DZF/DZFF
            AFFR=AFF*CPD(I,J,K+2)
         ELSE
            AFF=0.
            AFFR=0.
         ENDIF

         IF (K.GT.2) THEN
            ABB=-CBP*DZB/DZBB
            ABBR=ABB*CPD(I,J,K-2)
         ELSE
            ABB=0.
            ABBR=0.
         ENDIF

C *** MODIFICATION FOR DECK BOUNDARIES
         IF (NOD(I-1,J,K).NE.0) THEN
            AWW=0.0
            AWWR=0.0
         ENDIF

         IF (NOD(I+1,J,K).NE.0) THEN
            AEE=0.0
            AEER=0.0
         ENDIF

         IF (NOD(I,J-1,K).NE.0) THEN
            ASS=0.0
            ASSR=0.0
         ENDIF

         IF (NOD(I,J+1,K).NE.0) THEN
            ANN=0.0
            ANNR=0.0
         ENDIF

         IF (NOD(I,J,K-1).NE.0) THEN
            ABB=0.0
            ABBR=0.0
         ENDIF

         IF (NOD(I,J,K+1).NE.0) THEN
            AFF=0.0
            AFFR=0.0
```

```
         ENDIF

         AP(I,J,K)=AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)+AF(I,J,K)+
     &             AB(I,J,K)+AEE+AWW+ANN+ASS+AFF+ABB+CONDE1+CONDW1+
     &             CONDN1+CONDS1+CONDF1+CONDB1

         AE(I,J,K)=AE(I,J,K)+CONDE1
         AW(I,J,K)=AW(I,J,K)+CONDW1
         AN(I,J,K)=AN(I,J,K)+CONDN1
         AS(I,J,K)=AS(I,J,K)+CONDS1
         AF(I,J,K)=AF(I,J,K)+CONDF1
         AB(I,J,K)=AB(I,J,K)+CONDB1

         SP(I,J,K)=-ROD(I,J,K)*VOLDT
         SU(I,J,K)=-SP(I,J,K)*COD(I,J,K)+AEER+AWWR+ANNR+ASSR+AFFR+ABBR
  100 CONTINUE

C *** TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU

C *** Y DIRECTION
      DO 500 I=2,NI+3
      DO 500 K=2,NK+3
         SP(I,3   ,K)=SP(I,3   ,K)+AS(I,3   ,K)
         SP(I,NJ+2,K)=SP(I,NJ+2,K)+AN(I,NJ+2,K)
         AS(I,3   ,K)=0.
         AN(I,NJ+2,K)=0.
  500 CONTINUE

C *** X DIRECTION
      DO 600 J=2,NJ+3
      DO 600 K=2,NK+3
         SP(3   ,J,K)=SP(3   ,J,K)+AW(3   ,J,K)
         SP(NI+2,J,K)=SP(NI+2,J,K)+AE(NI+2,J,K)
         AW(3   ,J,K)=0.0
         AE(NI+2,J,K)=0.0
  600 CONTINUE

C *** Z DIRECTION
      DO 700 I=2,NI+3
      DO 700 J=2,NJ+3
         SP(I,J,3   )=SP(I,J,3   )+AB(I,J,3   )
         SP(I,J,NK+2)=SP(I,J,NK+2)+AF(I,J,NK+2)
         AB(I,J,3   )=0.
         AF(I,J,NK+2)=0.
  700 CONTINUE

C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
      DO 300 K=2,NK+3
      DO 300 J=2,NJ+3
      DO 300 I=2,NI+3
         AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
  300 CONTINUE

C *** VOLUMETRIC MASS SOURCE INPUT
      VOLT=0.0
      DO 113 I=2,NI+3
```

```
      DO 113 J=2,NJ+3
      DO 113 K=2,NK+3
         DXI =DXXC(I)
         DYJ =DYYC(J)
         DZK =DZZC(K)
         VOL =DXI*DYJ*DZK*H**3
         VOLT=VOLT+VOL
  113 CONTINUE

      DO 111 I=NHSZ(1,1),NHSZ(1,2)
      DO 111 J=NHSZ(2,1),NHSZ(2,2)
      DO 111 K=NHSZ(3,1),NHSZ(3,2)
         DXI =DXXC(I)
         DYJ =DYYC(J)
         DZK =DZZC(K)
         VOL =DXI*DYJ*DZK
         SU(I,J,K)=SU(I,J,K)+VOL*H/(U0*RHO0*VOLT)
  111 CONTINUE

C *** SOLVE FOR C
      CALL TRID (3,3,3,NI+2,NJ+2,NK+2,C)

C *** Z DIRECTION
      DO 74 I=1,NI+4
      DO 74 J=1,NJ+4
         C(I,J,2   )=C(I,J,3   )
         C(I,J,1   )=C(I,J,2   )
         C(I,J,NK+3)=C(I,J,NK+2)
         C(I,J,NK+4)=C(I,J,NK+3)
   74 CONTINUE

C *** Y DIRECTION
      DO 84 I=2,NI+3
      DO 84 K=2,NK+3
         C(I,NJ+3,K)=C(I,NJ+4,K)
         C(I,NJ+4,K)=C(I,NJ+3,K)
         C(I,2   ,K)=C(I,   3,K)
         C(I,1   ,K)=C(I,   2,K)
  84  CONTINUE

C *** X DIRECTION
      DO 80 J=1,NJ+4
      DO 80 K=1,NK+4
         C(2   ,J,K)=C(   3,J,K)
         C(1   ,J,K)=C(   2,J,K)
         C(NI+3,J,K)=C(NI+2,J,K)
         C(NI+4,J,K)=C(NI+3,J,K)
  80  CONTINUE

      RETURN
      END
```

****

****

```
      SUBROUTINE CALP
*******************************************************************************
*CALCULATES NODE PRESSURES

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL12/NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
     &            JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
      COMMON/BL23/RMS(20),NMS,IMSB(20),NMSI(20),JMSB(20),NMSJ(20),
     &            KMSB(20),NMSK(20)
      COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
     &            COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
     &            WOD(25,25,15)
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
     &            SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
     &            DV(25,25,15),DW(25,25,15)
      COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
     &            AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
     &            SU(25,25,15),RI(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

C *** CALCULATE COEFFICIENTS
      DO 100 K=2,NK+3
      DO 100 J=2,NJ+3
      DO 100 I=2,NI+3

      IF (NOD(I,J,K).EQ.1) GOTO 100

C *** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
         DXP1=DXXC(I+1)
         DXI =DXXC(I)
         DXM1=DXXC(I-1)

         DYP1=DYYC(J+1)
         DYJ =DYYC(J)
         DYM1=DYYC(J-1)

         DZP1=DZZC(K+1)
         DZK =DZZC(K)
         DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
         DXN=DXXC(I)
         DXS=DXXC(I)
         DXF=DXXC(I)
         DXB=DXXC(I)

         DYF=DYYC(J)
         DYB=DYYC(J)
```

```
          DYE=DYYC(J)
          DYW=DYYC(J)

          DZE=DZZC(K)
          DZW=DZZC(K)
          DZN=DZZC(K)
          DZS=DZZC(K)

C *** DEFINE AREA OF THE CONTROL VOLUME
          DXYF=DXF*DYF
          DXYB=DXB*DYB
          DYZE=DYE*DZE
          DYZW=DYW*DZW
          DZXN=DZN*DXN
          DZXS=DZS*DXS

          VOL=DXI*DYJ*DZK
          VOLDT=VOL/DTIME

C *** DENSITY AT THE SURFACES
          RN=(R(I,J,K)*DYP1+R(I  ,J+1,K  )*DYJ)/(DYP1+DYJ)
          RS=(R(I,J,K)*DYM1+R(I  ,J-1,K  )*DYJ)/(DYM1+DYJ)
          RE=(R(I,J,K)*DXP1+R(I+1,J  ,K  )*DXI)/(DXP1+DXI)
          RW=(R(I,J,K)*DXM1+R(I-1,J  ,K  )*DXI)/(DXM1+DXI)
          RF=(R(I,J,K)*DZP1+R(I  ,J  ,K+1)*DZK)/(DZP1+DZK)
          RB=(R(I,J,K)*DZM1+R(I  ,J  ,K-1)*DZK)/(DZM1+DZK)

          AN(I,J,K)=RN*DZXN*DV(I  ,J+1,K  )
          AS(I,J,K)=RS*DZXS*DV(I  ,J  ,K  )
          AE(I,J,K)=RE*DYZE*DU(I+1,J  ,K  )
          AW(I,J,K)=RW*DYZW*DU(I  ,J  ,K  )
          AF(I,J,K)=RF*DXYF*DW(I  ,J  ,K+1)
          AB(I,J,K)=RB*DXYB*DW(I  ,J  ,K  )

          CN=RN*V(I  ,J+1,K  )*DZXN
          CS=RS*V(I  ,J  ,K  )*DZXS
          CE=RE*U(I+1,J  ,K  )*DYZE
          CW=RW*U(I  ,J  ,K  )*DYZW
          CF=RF*W(I  ,J  ,K+1)*DXYF
          CB=RB*W(I  ,J  ,K  )*DXYB

          SMP(I,J,K)=-(R(I,J,K)-ROD(I,J,K))*VOLDT-CE+CW-CN+CS-CF+CB
          SU (I,J,K)= SMP(I,J,K)
          SP (I,J,K)= 0.
  100 CONTINUE

C *** CONSIDER THE MASS SOURCE INPUT INTO THE CONTROL VOLUME
      IF (NMS.GE.1) THEN
         DO 150 M=1,NMS
            IB=IMSB(M)
            IE=IB+NMSI(M)-1
            JB=JMSB(M)
            JE=JB+NMSJ(M)-1
            KB=KMSB(M)
            KE=KB+NMSK(M)-1
            DO 160 I=IB,IE-1
```

```
            DO 160 J=JB,JE-1
            DO 160 K=KB,KE-1
               SU(I,J,K)=SU(I,J,K)+RMS(M)
  160       CONTINUE
  150    CONTINUE
      ENDIF

C *** TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU

C *** X DIRECTION
      DO 500 K=2,NK+3
      DO 500 I=2,NI+3
         AS(I,2   ,K)=0.
         AN(I,NJ+3,K)=0.
  500 CONTINUE

C *** Y DIRECTION
      DO 501 K=2,NK+3
      DO 501 J=2,NJ+3
         AW(2   ,J,K)=0.
         AE(NI+3,J,K)=0.
  501 CONTINUE

C *** Z -DIRECTION
      DO 502 I=2,NI+3
      DO 502 J=2,NJ+3
         AB(I,J,2   )=0.
         AF(I,J,NK+3)=0.
 502  CONTINUE

C *** MODIFICATION FOR DECK BOUNDARIES
      IF (NCHIP.EQ.0) GOTO 110
      DO 101 N=1,NCHIP
         IB  =ICHPB(N)
         IE  =IB+NCHPI(N)-1
         JB  =JCHPB(N)
         JE  =JB+NCHPJ(N)-1
         KB  =KCHPB(N)
         KE  =KB+NCHPK(N)-1

         DO 102 J=JB,JE-1
         DO 102 K=KB,KE-1
            AE(IB-1,J,K)=0.0
            AW(IE  ,J,K)=0.0
  102    CONTINUE

         DO 103 I=IB,IE-1
         DO 103 K=KB,KE-1
            AN(I,JB-1,K)=0.0
            AS(I,JE  ,K)=0.0
  103    CONTINUE

         DO 106 I=IB,IE-1
         DO 106 J=JB,JE-1
            AF(I,J,KB-1)=0.0
            AB(I,J,KE  )=0.0
```

```
  106    CONTINUE

C *** FOR THE CELLS INSIDE OF THE DECKS
         DO 104 I=IB,IE-1
         DO 104 J=JB,JE-1
         DO 104 K=KB,KE-1
            SP(I,J,K)=-1.0E2
            AW(I,J,K)=0.
            AE(I,J,K)=0.
            AS(I,J,K)=0.
            AN(I,J,K)=0.
            AB(I,J,K)=0.
            AF(I,J,K)=0.
            SU(I,J,K)=0.
  104    CONTINUE
  101 CONTINUE

C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
  110 DO 300 I=2,NI+3
      DO 300 J=2,NJ+3
      DO 300 K=2,NK+3
         AP(I,J,K)=AN(I,J,K)+AS(I,J,K)+AE(I,J,K)+AW(I,J,K)-SP(I,J,K)
     &             +AF(I,J,K)+AB(I,J,K)
  300 CONTINUE

C *** SOLUTION OF FINITE DIFFERENCE EQUATION
      CALL TRID  (3,3,3,NI+2,NJ+2,NK+2,PP)

C *** CORRECTION FOR VELOCITY U
      DO 600 I=3,NI+3
      DO 600 J=2,NJ+3
      DO 600 K=2,NK+3
         U(I,J,K)=U(I,J,K)+DU(I,J,K)*(PP(I-1,J,K)-PP(I,J,K))
  600 CONTINUE

C *** CORRECTION FOR VELOCITY V
      DO 603 J=3,NJ+3
      DO 603 K=2,NK+3
      DO 603 I=2,NI+3
         V(I,J,K)=V(I,J,K)+DV(I,J,K)*(PP(I,J-1,K)-PP(I,J,K))
  603 CONTINUE

C *** CORRECTION FOR VELOCITY W
      DO 604 K=3,NK+3
      DO 604 I=2,NI+3
      DO 604 J=2,NJ+3
         W(I,J,K)=W(I,J,K)+DW(I,J,K)*(PP(I,J,K-1)-PP(I,J,K))
 604  CONTINUE

C *** CORRECTION FOR PRESSURE P
      DO 606 J=1,NJ+4
      DO 606 I=1,NI+4
      DO 606 K=1,NK+4
         P (I,J,K)=P(I,J,K)+PP(I,J,K)
         PP(I,J,K)=0.
  606 CONTINUE
```

```
C *** RESET THE VELOCITY INSIDE OF DECK
      IF (NCHIP.EQ.0) GOTO 121
      DO 120 N=1,NCHIP
         IB=ICHPB(N)
         IE=IB+NCHPI(N)-1
         JB=JCHPB(N)
         JE=JB+NCHPJ(N)-1
         KB=KCHPB(N)
         KE=KB+NCHPK(N)-1
         DO 109 I=IB,IE
         DO 109 J=JB,JE-1
         DO 109 K=KB,KE-1
            U(I,J,K)=0.0
  109    CONTINUE

         DO 118 I=IB,IE-1
         DO 118 J=JB,JE
         DO 118 K=KB,KE-1
            V(I,J,K)=0.0
  118    CONTINUE

         DO 119 I=IB,IE-1
         DO 119 J=JB,JE-1
         DO 119 K=KB,KE
            W(I,J,K)=WFAN(N)
  119    CONTINUE
  120 CONTINUE

C *** RECALCULATE THE ERROR SOURCE AFTER CORRECTIONS OF U, V, P
  121 SORSUM=0.
      RESORM(ITER)=0.
      DO 700 J=2,NJ+3
      DO 700 I=2,NI+3
      DO 700 K=2,NK+3
         IF (NOD(I,J,K).NE.1) THEN

C *** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
            DXP1=DXXC(I+1)
            DXI =DXXC(I)
            DXM1=DXXC(I-1)

            DYP1=DYYC(J+1)
            DYJ =DYYC(J)
            DYM1=DYYC(J-1)

            DZP1=DZZC(K+1)
            DZK =DZZC(K)
            DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
            DXN=DXXC(I)
            DXS=DXXC(I)
            DXF=DXXC(I)
            DXB=DXXC(I)
```

```
            DYF=DYYC(J)
            DYB=DYYC(J)
            DYE=DYYC(J)
            DYW=DYYC(J)

            DZE=DZZC(K)
            DZW=DZZC(K)
            DZN=DZZC(K)
            DZS=DZZC(K)

C *** DEFINE AREA OF THE CONTROL VOLUME
            DXYF=DXF*DYF
            DXYB=DXB*DYB
            DYZE=DYE*DZE
            DYZW=DYW*DZW
            DZXN=DZN*DXN
            DZXS=DZS*DXS

            VOL=DXI*DYJ*DZK
            VOLDT=VOL/DTIME

C *** CALCULATE DENSITY
            RN=(R(I,J,K)*DYP1+R(I  ,J+1,K  )*DYJ)/(DYP1+DYJ)
            RS=(R(I,J,K)*DYM1+R(I  ,J-1,K  )*DYJ)/(DYM1+DYJ)
            RE=(R(I,J,K)*DXP1+R(I+1,J  ,K  )*DXI)/(DXP1+DXI)
            RW=(R(I,J,K)*DXM1+R(I-1,J  ,K  )*DXI)/(DXM1+DXI)
            RF=(R(I,J,K)*DZP1+R(I  ,J  ,K+1)*DZK)/(DZP1+DZK)
            RB=(R(I,J,K)*DZM1+R(I  ,J  ,K-1)*DZK)/(DZM1+DZK)

            CN=RN*V(I  ,J+1,K  )*DZXN
            CS=RS*V(I  ,J  ,K  )*DZXS
            CE=RE*U(I+1,J  ,K  )*DYZE
            CW=RW*U(I  ,J  ,K  )*DYZW
            CF=RF*W(I  ,J  ,K+1)*DXYF
            CB=RB*W(I  ,J  ,K  )*DXYB
            SMP(I,J,K)=(ROD(I,J,K)-R(I,J,K))*VOLDT-CE+CW-CN+CS-CF+CB

C *** SORSUM IS ACTUAL MASS INCREASE OR DECREASE FROM CONTINUITY
C *** EQUATION, THIS WILL BE COMPARED TO MASS SOURCE

C *** CONSIDER THE MASS SOURCE INPUT INTO THE CONTROL VOLUME
            IF (NMS.GT.0) THEN
               DO 250 M=1,NMS
                  IB=IMSB(M)
                  IE=IB+NMSI(M)-1
                  JB=JMSB(M)
                  JE=JB+NMSJ(M)-1
                  KB=KMSB(M)
                  KE=KB+NMSK(M)-1
                  DO 260 II=IB,IE-1
                  DO 260 JJ=JB,JE-1
                  DO 260 KK=KB,KE-1
                     IF ((II.EQ.I).AND.(JJ.EQ.J).AND.(KK.EQ.K)) THEN
                        SMP(I,J,K)=SMP(I,J,K)+RMS(M)
                     ENDIF
  260             CONTINUE
```

```
  250          CONTINUE
            ENDIF
            SORSUM=SORSUM+SMP(I,J,K)

C *** RESORM IS SUM OF THE ABSOLUTE VALUE OF SMP(I,J,K)
            RESORM(ITER)=RESORM(ITER)+ABS(SMP(I,J,K))
         ENDIF

  700 CONTINUE
      RETURN
      END

*******************************************************************************
*******************************************************************************
      SUBROUTINE CALQ
*******************************************************************************
*
*VARIABLES:
*  BR     = MAXIMUM BURN RATE (LBM/SEC)
*  F      = MAXIMUM FUEL AVAILABLE (LBM)
*  FR     = TOTAL FUEL REMAINING (LBM)
*  H      = REFERENCE LENGTH (FT)
*  HC     = HEAT OF COMBUSTION (BTU/LBM)
*  HSTART= FIRE START TIME (SECONDS)
*  Q      = TOTAL HEAT INPUT (BTU/SEC)
*  TIME   = NONDIMENSIONAL FIRE TIME
*  U0     = REFERENCE VELOCITY (FT/SEC)
*  XTIME = FIRE TIME (SECONDS)
*
*******************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL2/X,Y,H,TFLR,TWAL
      COMMON/BL3/F,FR,HSTART
      COMMON/BL12/NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL16/U0,UGRT,BUOY,CP0,PRT,COND0,VIS0,RHO0,
     &            TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT

      XTIME=TIME*H/U0
      HC=2600.0
      BR=   0.01

C *** CALCULATE HEAT RELEASE RATE (Q) IN BTU/SEC
C *** NOTE:  THESE ALGORITHMS ASSUME A LINEAR INCREASE IN BOTH
C ***        HEAT RELEASE AND FUEL CONSUMPTION OVER THE FIRST
C ***        TWO SECONDS OF FIRE TIME, AFTER WHICH BOTH ARE AT
C ***        MAXIMUM
      IF(XTIME.LT.HSTART) THEN
         Q=0.0
         FR=F
      ELSEIF(XTIME.GE.HSTART.AND.(XTIME-HSTART).LE.2.0) THEN
         IF(FR.LE.0.0) THEN
            Q =0.0
            FR=0.0
         ELSE
```

```
            Q =HC*BR*(XTIME-HSTART)/2.
            FR=F-BR*(XTIME-HSTART)**2/2.
         ENDIF
      ELSEIF((XTIME-HSTART).GT.2.0) THEN
         IF(FR.LE.0.0) THEN
            Q =0.0
            FR=0.0
         ELSE
            Q =HC*BR
            FR=F-BR*(XTIME-HSTART)
         ENDIF
      ENDIF

C *** TAKE RADIATION HEAT FLUX INTO ACCOUNT
      Q=Q-QR
      IF (Q.LE.0.0) Q=0.

      RETURN
      END

*************************************************************************
*************************************************************************
      SUBROUTINE CALT
*************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL2/X,Y,H,TFLR,TWAL
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL14/HCOEF,CNT,ABTURB,BTURB,VISL,VISMAX
      COMMON/BL16/U0,UGRT,BUOY,CP0,PRT,COND0,VIS0,RHO0,
     &            TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
      COMMON/BL23/RMS(20),NMS,IMSB(20),NMSI(20),JMSB(20),NMSJ(20),
     &            KMSB(20),NMSK(20)
      COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
     &            COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
     &            WOD(25,25,15)
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
     &            CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
     &            WPD(25,25,15)
      COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
     &            SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
     &            DV(25,25,15),DW(25,25,15)
      COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
     &            AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
     &            SU(25,25,15),RI(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
      COMMON/BL39/ALEW,CONSRA,QSIN,QSWER,QSWAL,QSAIR,QSFAN
      COMMON/BL43/QSCONF,QSCONB,QSCONE,QSCONW,QSCONN,QSCONS,
     &            QSRADF,QSRADB,QSRADE,QSRADW,QSRADN,QSRADS,
     &            WAIR,WWAL,WINS,WERR,WWFAN
```

```
C *** NONDIMENSIONAL REFERENCE TEMPERATURE
      TINF=TA/TA

C *** CALCULATE COEFFICIENTS
      DO 100 K=2,NK+3
      DO 100 J=2,NJ+3
      DO 100 I=2,NI+3

C *** CENTRAL LENGTH OF THE TEMPERTURE CONTROL VOLUME
         DXP1=DXXC(I+1)
         DXI =DXXC(I)
         DXM1=DXXC(I-1)

         DYP1=DYYC(J+1)
         DYJ =DYYC(J)
         DYM1=DYYC(J-1)

         DZP1=DZZC(K+1)
         DZK =DZZC(K)
         DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
         DXN=DXXC(I)
         DXS=DXXC(I)
         DXF=DXXC(I)
         DXB=DXXC(I)

         DYF=DYYC(J)
         DYB=DYYC(J)
         DYE=DYYC(J)
         DYW=DYYC(J)

         DZE=DZZC(K)
         DZW=DZZC(K)
         DZN=DZZC(K)
         DZS=DZZC(K)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR TEMPERATURE
         DXEE=DXXS(I+2)
         DXE =DXXS(I+1)
         DXW =DXXS(I)
         DXWW=DXXS(I-1)

         DYNN=DYYS(J+2)
         DYN =DYYS(J+1)
         DYS =DYYS(J)
         DYSS=DYYS(J-1)

         DZFF=DZZS(K+2)
         DZF =DZZS(K+1)
         DZB =DZZS(K)
         DZBB=DZZS(K-1)

C *** DEFINE THE AREA OF THE CONTROL VOLUME
         DXYF=DXF*DYF
```

```
          DXYB=DXB*DYB
          DYZE=DYE*DZE
          DYZW=DYW*DZW
          DZXN=DZN*DXN
          DZXS=DZS*DXS

          VOL=DXI*DYJ*DZK
          VOLDT=VOL/DTIME

C *** FOR CONDUCTION
          ZXOYN=DZXN/DYN
          ZXOYS=DZXS/DYS
          XYOZF=DXYF/DZF
          XYOZB=DXYB/DZB
          YZOXE=DYZE/DXE
          YZOXW=DYZW/DXW

C *** DENSITY AT THE SURFACES
          GN=(R(I,J,K)*DYP1+R(I,J+1,K)*DYJ)/(DYP1+DYJ)
          GS=(R(I,J,K)*DYM1+R(I,J-1,K)*DYJ)/(DYM1+DYJ)
          GE=(R(I,J,K)*DXP1+R(I+1,J,K)*DXI)/(DXP1+DXI)
          GW=(R(I,J,K)*DXM1+R(I-1,J,K)*DXI)/(DXM1+DXI)
          GF=(R(I,J,K)*DZP1+R(I,J,K+1)*DZK)/(DZP1+DZK)
          GB=(R(I,J,K)*DZM1+R(I,J,K-1)*DZK)/(DZM1+DZK)

C *** THE MASS FLUX RATE THROUGH THE SURFACES
          CN=GN*V(I,J+1,K)*DZXN
          CS=GS*V(I,J  ,K)*DZXS
          CE=GE*U(I+1,J,K)*DYZE
          CW=GW*U(I  ,J,K)*DYZW
          CF=GF*W(I,J,K+1)*DXYF
          CB=GB*W(I,J,K  )*DXYB

C *** CONDUCTIVITY AT THE SURFACES
          CONDN=(DYP1+DYJ)*COND(I,J,K)*COND(I,J+1,K)*DYJ*DYP1/
     &          (DYJ*COND(I,J,K)+DYP1*COND(I,J+1,K))
          CONDS=(DYM1+DYJ)*COND(I,J,K)*COND(I,J-1,K)*DYJ*DYM1/
     &          (DYJ*COND(I,J,K)+DYM1*COND(I,J-1,K))
          CONDE=(DXP1+DXI)*COND(I,J,K)*COND(I+1,J,K)*DXI*DXP1/
     &          (DXI*COND(I,J,K)+DXP1*COND(I+1,J,K))
          CONDW=(DXM1+DXI)*COND(I,J,K)*COND(I-1,J,K)*DXI*DXM1/
     &          (DXI*COND(I,J,K)+DXM1*COND(I-1,J,K))
          CONDF=(DZP1+DZK)*COND(I,J,K)*COND(I,J,K+1)*DZK*DZP1/
     &          (DZK*COND(I,J,K)+DZP1*COND(I,J,K+1))
          CONDB=(DZM1+DZK)*COND(I,J,K)*COND(I,J,K-1)*DZK*DZM1/
     &          (DZK*COND(I,J,K)+DZM1*COND(I,J,K-1))

C *** CONDUCTION COMPONENT
          CONDN1=ZXOYN*CONDN
          CONDS1=ZXOYS*CONDS
          CONDE1=YZOXE*CONDE
          CONDW1=YZOXW*CONDW
          CONDF1=XYOZF*CONDF
          CONDB1=XYOZB*CONDB

C *** QUICK SCHEME
```

```
      CEP=(ABS(CE)+CE)*DXP1*DXI/(DXE*(DXE+DXW )*8.)
      CEM=(ABS(CE)-CE)*DXP1*DXI/(DXE*(DXE+DXEE)*8.)
      CWP=(ABS(CW)+CW)*DXM1*DXI/(DXW*(DXW+DXWW)*8.)
      CWM=(ABS(CW)-CW)*DXM1*DXI/(DXW*(DXW+DXE )*8.)

      CNP=(ABS(CN)+CN)*DYP1*DYJ/(DYN*(DYN+DYS )*8.)
      CNM=(ABS(CN)-CN)*DYP1*DYJ/(DYN*(DYN+DYNN)*8.)
      CSP=(ABS(CS)+CS)*DYM1*DYJ/(DYS*(DYS+DYSS)*8.)
      CSM=(ABS(CS)-CS)*DYM1*DYJ/(DYS*(DYS+DYN )*8.)

      CFP=(ABS(CF)+CF)*DZP1*DZK/(DZF*(DZF+DZB )*8.)
      CFM=(ABS(CF)-CF)*DZP1*DZK/(DZF*(DZF+DZFF)*8.)
      CBP=(ABS(CB)+CB)*DZM1*DZK/(DZB*(DZB+DZBB)*8.)
      CBM=(ABS(CB)-CB)*DZM1*DZK/(DZB*(DZB+DZF )*8.)

      AE(I,J,K)=-.5*CE*DXI/DXE+CEP+CEM*(1.+DXE/DXEE)+CWM*DXW/DXE
      AW(I,J,K)= .5*CW*DXI/DXW+CWM+CWP*(1.+DXW/DXWW)+CEP*DXE/DXW
      AN(I,J,K)=-.5*CN*DYJ/DYN+CNP+CNM*(1.+DYN/DYNN)+CSM*DYS/DYN
      AS(I,J,K)= .5*CS*DYJ/DYS+CSM+CSP*(1.+DYS/DYSS)+CNP*DYN/DYS
      AF(I,J,K)=-.5*CF*DZK/DZF+CFP+CFM*(1.+DZF/DZFF)+CBM*DZB/DZF
      AB(I,J,K)= .5*CB*DZK/DZB+CBM+CBP*(1.+DZB/DZBB)+CFP*DZF/DZB

C *** BOUNDARY CONSIDERATIONS
      IF (I.LT.NI+3) THEN
         AEE=-CEM*DXE/DXEE
         AEER=AEE*TPD(I+2,J,K)*CPM(I+2,J,K)
      ELSE
         AEE=0.
         AEER=0.
      ENDIF

      IF (I.GT.2) THEN
         AWW=-CWP*DXW/DXWW
         AWWR=AWW*TPD(I-2,J,K)*CPM(I-2,J,K)
      ELSE
         AWW=0.
         AWWR=0.
      ENDIF

      IF (J.LT.NJ+3) THEN
         ANN=-CNM*DYN/DYNN
         ANNR=ANN*TPD(I,J+2,K)*CPM(I,J+2,K)
      ELSE
         ANN=0.
         ANNR=0.
      ENDIF

      IF (J.GT.2) THEN
         ASS=-CSP*DYS/DYSS
         ASSR=ASS*TPD(I,J-2,K)*CPM(I,J-2,K)
      ELSE
         ASS=0.
         ASSR=0.
      ENDIF

      IF (K.LT.NK+3) THEN
```

```
            AFF=-CFM*DZF/DZFF
            AFFR=AFF*TPD(I,J,K+2)*CPM(I,J,K+2)
         ELSE
            AFF=0.
            AFFR=0.
         ENDIF

         IF (K.GT.2) THEN
            ABB=-CBP*DZB/DZBB
            ABBR=ABB*TPD(I,J,K-2)*CPM(I,J,K-2)
         ELSE
            ABB=0.
            ABBR=0.
         ENDIF

C *** MODIFICATION FOR DECK BOUNDARIES
         IF (NOD(I-1,J,K).NE.0) THEN
            AWW=0.0
            AWWR=0.0
         ENDIF

         IF (NOD(I+1,J,K).NE.0) THEN
            AEE=0.0
            AEER=0.0
         ENDIF

         IF (NOD(I,J-1,K).NE.0) THEN
            ASS=0.0
            ASSR=0.0
         ENDIF

         IF (NOD(I,J+1,K).NE.0) THEN
            ANN=0.0
            ANNR=0.0
         ENDIF

         IF (NOD(I,J,K-1).NE.0) THEN
            ABB=0.0
            ABBR=0.0
         ENDIF

         IF (NOD(I,J,K+1).NE.0) THEN
            AFF=0.0
            AFFR=0.0
         ENDIF

         AP(I,J,K)=(AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)+AF(I,J,K)+
     &              AB(I,J,K)+AEE+AWW+ANN+ASS+AFF+ABB)*CPM(I,J,K)+
     &              CONDE1+CONDW1+CONDN1+CONDS1+CONDF1+CONDB1
         AE(I,J,K)=AE(I,J,K)*CPM(I+1,J  ,K  )+CONDE1
         AW(I,J,K)=AW(I,J,K)*CPM(I-1,J  ,K  )+CONDW1
         AN(I,J,K)=AN(I,J,K)*CPM(I  ,J+1,K  )+CONDN1
         AS(I,J,K)=AS(I,J,K)*CPM(I  ,J-1,K  )+CONDS1
         AF(I,J,K)=AF(I,J,K)*CPM(I  ,J  ,K+1)+CONDF1
         AB(I,J,K)=AB(I,J,K)*CPM(I  ,J  ,K-1)+CONDB1
```

```
         SP(I,J,K)=-ROD(I,J,K)*VOLDT*CPM(I,J,K)
         SU(I,J,K)=-SP(I,J,K)*TOD(I,J,K)+AEER+AWWR+ANNR+ASSR+AFFR+ABBR
  100 CONTINUE

C *** TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU

C *** Y-DIRECTION
      DO 500 I=3,NI+2
      DO 500 K=3,NK+2
         SU(I,3   ,K)=SU(I,3   ,K)+AS(I,3   ,K)*T(I,2   ,K)
         SU(I,NJ+2,K)=SU(I,NJ+2,K)+AN(I,NJ+2,K)*T(I,NJ+3,K)
         AS(I,3   ,K)=0.
         AN(I,NJ+2,K)=0.
  500 CONTINUE

C *** X-DIRECTION
      DO 600 J=3,NJ+2
      DO 600 K=3,NK+2
         SU(3   ,J,K)=SU(3   ,J,K)+AW(3   ,J,K)*T(2   ,J,K)
         SU(NI+2,J,K)=SU(NI+2,J,K)+AE(NI+2,J,K)*T(NI+3,J,K)
         AW(3   ,J,K)=0.0
         AE(NI+2,J,K)=0.0
  600 CONTINUE

C *** Z-DIRECTION
      DO 700 I=3,NI+2
      DO 700 J=3,NJ+2
      SU(I,J,3   )=SU(I,J,3   )+AB(I,J,3   )*T(I,J,2   )
      SU(I,J,NK+2)=SU(I,J,NK+2)+AF(I,J,NK+2)*T(I,J,NK+3)
      AB(I,J,3   )=0.
      AF(I,J,NK+2)=0.
  700 CONTINUE

C *** CONSIDER THE MASS SOURCE INPUT TO THE CONTROL VOLUME
      IF (NMS.GE.1) THEN
         DO 150 M=1,NMS
            IB=IMSB(M)
            IE=IB+NMSI(M)-1
            JB=JMSB(M)
            JE=JB+NMSJ(M)-1
            KB=KMSB(M)
            KE=KB+NMSK(M)-1
            DO 160 I=IB,IE-1
            DO 160 J=JB,JE-1
            DO 160 K=KB,KE-1
               IF (RMS(M).GE.0.0) THEN
                  RMSCPT=RMS(M)*1.0*CPM(I,J,K)
               ELSE
                  RMSCPT=RMS(M)*T(I,J,K)*CPM(I,J,K)*R(I,J,K)
               ENDIF
  160       CONTINUE
  150    CONTINUE
      ENDIF

C *** CONSIDER THE RADIATION HEAT FLUX FROM THE FIRE TO THE BLOCK
      CALL RADHT (2)
```

```
C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
      DO 300 K=3,NK+2
      DO 300 J=3,NJ+2
      DO 300 I=3,NI+2
         AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
  300 CONTINUE

C *** VOLUME HEAT SOURCE INPUT

C *** CALCULATE THE TOTAL VOLUME OCCUPIED BY HEAT SOURCE
C *** DISTRIBUTE ENERGY INTO EACH CONTROL VOLUME
C *** QQQ/H**3    DIMENSIONLESS HEAT SOURCE

      VOLT=0.0
      DO 113 I=NHSZ(1,1),NHSZ(1,2)
      DO 113 J=NHSZ(2,1),NHSZ(2,2)
      DO 113 K=NHSZ(3,1),NHSZ(3,2)
         QQQ=Q*H/(U0*CP0*RHO0*TA)
         VOL=DXXC(I)*DYYC(J)*DZZC(K)
         VOLT=VOLT+VOL*H**3
         SU(I,J,K)=SU(I,J,K)+VOL*QQQ/VOLT
  113 CONTINUE

C *** SOLVE FOR T
      CALL TRID (3,3,3,NI+2,NJ+2,NK+2,T)

      DO 2001 I=1,NI+4
      DO 2001 J=1,NJ+4
      DO 2001 K=1,NK+4
         IF(T(I,J,K).LT.TCOOL) T(I,J,K)=TCOOL
 2001 CONTINUE

C *** CALCULATE RADIATION HEAT TRANSFER

C *** HERE SU(I,J,K) IS USED TO STORE THE RADIATIVE HEAT FLUX
      DO 75 I=1,NI+4
      DO 75 J=1,NJ+4
      DO 75 K=1,NK+4
         SU(I,J,K)=0.
   75 CONTINUE

C *** CONSIDER THE RADIATION HEAT FLUX FROM THE FIRE TO THE WALL
      CALL RADHT (1)

C *** SUMMATION OF CONDUCTION HEAT FLUX AND RADIATION HEAT FLUX TO WALLS
      QSCONF=0.
      QSCONB=0.
      QSCONE=0.
      QSCONW=0.
      QSCONN=0.
      QSCONS=0.

      QSRADF=0.
      QSRADB=0.
      QSRADE=0.
```

```
      QSRADW=0.
      QSRADN=0.
      QSRADS=0.

C *** CALCULATE CONDUCTION, RADIATION & TEMPERATURE ON THE SOLID WALLS
      DO 74 I=3,NI+2
      DO 74 J=3,NJ+2

C *** ON THE BACK WALL
         DZK =DZZC(2)
         DZP1=DZZC(3)
         DXI =DXXC(I)
         DYJ =DYYC(J)
         DXY =DXI*DYJ
         VOL =DXY*DZK
         CONDF=(DZP1+DZK)*DZK*DZP1*COND(I,J,2)*COND(I,J,3)/
     &        (DZK*COND(I,J,2)+DZP1*COND(I,J,3))
         QCONF=DXY*CONDF*(T(I,J,3)-T(I,J,2))*2.0/(DZP1+DZK)
         QCONB=DXY*COND(I,J,2)*(T(I,J,1)-T(I,J,2))*2.0/DZK
         QRADB=SU(I,J,2)

         T(I,J,2)=TOD(I,J,2)+DTIME*(QCONF+QCONB+QRADB)/(VOL*CPM(I,J,2))
         T(I,J,1)=(2.*COND(I,J,2)*T(I,J,2)+HCOEF*TINF*DZK)/
     &            (HCOEF*DZK+2.*COND(I,J,2))

         QSCONB=QSCONB+QCONF
         QSRADB=QSRADB+QRADB

C *** ON THE FRONT WALL
         DZK =DZZC(NK+3)
         DZM1=DZZC(NK+2)
         DXI =DXXC(I)
         DYJ =DYYC(J)
         DXY =DXI*DYJ
         VOL =DXY*DZK
         CONDB=(DZM1+DZK)*DZK*DZM1*COND(I,J,NK+3)*COND(I,J,NK+2)/
     &        (DZK*COND(I,J,NK+3)+DZM1*COND(I,J,NK+2))
         QCONB=DXY*CONDB*(T(I,J,NK+2)-T(I,J,NK+3))*2.0/(DZK+DZM1)
         QCONF=DXY*COND(I,J,NK+3)*(T(I,J,NK+4)-T(I,J,NK+3))*2.0/DZK
         QRADF=SU(I,J,NK+3)

         T(I,J,NK+3)=TOD(I,J,NK+3)+DTIME*(QCONB+QCONF+QRADF)/
     &               (VOL*CPM(I,J,NK+3))
         T(I,J,NK+4)=(2.0*COND(I,J,NK+3)*T(I,J,NK+3)+HCOEF*TINF*DZK)/
     &               (HCOEF*DZK+2.0*COND(I,J,NK+3))

         QSCONF=QSCONF+QCONB
         QSRADF=QSRADF+QRADF

   74 CONTINUE

      DO 84 I=3,NI+2
      DO 84 K=3,NK+2

C *** ON THE SOUTH WALL
         DYJ =DYYC(2)
```

```
          DYP1=DYYC(3)
          DXI =DXXC(I)
          DZK =DZZC(K)
          DZX =DZK*DXI
          VOL =DZX*DYJ
          CONDN=(DYP1+DYJ)*DYJ*DYP1*COND(I,2,K)*COND(I,3,K)/
     &          (DYJ*COND(I,2,K)+DYP1*COND(I,3,K))
          QCONN=DZX*CONDN*(T(I,3,K)-T(I,2,K))*2.0/(DYP1+DYJ)
          QCONS=DZX*COND(I,2,K)*(T(I,1,K)-T(I,2,K))*2.0/DYJ
          QRADS=SU(I,2,K)
          T(I,2,K)=TOD(I,2,K)+DTIME*(QCONN+QCONS+QRADS)/
     &             (VOL*CPM(I,2,K))
          T(I,1,K)=(2.0*COND(I,2 ,K)*T(I,2 ,K)+HCOEF*TINF*DYJ)/
     &             (HCOEF*DYJ+2.0*COND(I,2 ,K))

          QSCONS=QSCONS+QCONN
          QSRADS=QSRADS+QRADS

C *** ON THE NORTH WALL
          DYJ =DYYC(NJ+3)
          DYM1=DYYC(NJ+2)
          DXI =DXXC(I)
          DZK =DZZC(K)
          DZX =DZK*DXI
          VOL =DZX*DYJ
          CONDS=(DYM1+DYJ)*DYJ*DYM1*COND(I,NJ+3,K)*COND(I,NJ+2,K)/
     &          (DYJ*COND(I,NJ+3,K)+DYM1*COND(I,NJ+2,K))
          QCONS=DZX*CONDS*(T(I,NJ+2,K)-T(I,NJ+3,K))*2.0/(DYM1+DYJ)
          QCONN=DZX*COND(I,NJ+3,K)*(T(I,NJ+4,K)-T(I,NJ+3,K))*2.0/DYJ
          QRADN=SU(I,NJ+3,K)
          T(I,NJ+3,K)=TOD(I,NJ+3,K)+DTIME*(QCONS+QCONN+QRADN)/
     &                (VOL*CPM(I,NJ+3,K))
          T(I,NJ+4,K)=(2.0*COND(I,NJ+3,K)*T(I,NJ+3,K)+HCOEF*TINF*DYJ)/
     &                (HCOEF*DYJ+2.0*COND(I,NJ+3,K))

          QSCONN=QSCONN+QCONS
          QSRADN=QSRADN+QRADN
   84  CONTINUE

       DO 80 J=3,NJ+2
       DO 80 K=3,NK+2

C *** ON THE WEST WALL
          DXI =DXXC(2)
          DXP1=DXXC(3)
          DYJ =DYYC(J)
          DZK =DZZC(K)
          DYZ =DYJ*DZK
          VOL =DYZ*DXI
          CONDE=(DXP1+DXI)*DXI*DXP1*COND(2,J,K)*COND(3,J,K)/
     &          (DXI*COND(2,J,K)+DXP1*COND(3,J,K))
          QCONE=DYZ*CONDE      *(T(3,J,K)-T(2,J,K))*2.0/(DXI+DXP1)
          QCONW=DYZ*COND(2,J,K)*(T(1,J,K)-T(2,J,K))*2.0/DXI
          QRADW=SU(2,J,K)
          T(2,J,K)=TOD(2,J,K)+DTIME*(QCONE+QCONW+QRADW)/(VOL*CPM(2,J,K))
          T(1,J,K)=(2.0*COND(2 ,J,K)*T(2 ,J,K)+HCOEF*TINF*DXI)/
```

```
     &            (HCOEF*DXI+2.0*COND(2,J,K))

          QSCONW=QSCONW+QCONE
          QSRADW=QSRADW+QRADW

C *** ON THE EAST WALL
          DXI =DXXC(NI+3)
          DXM1=DXXC(NI+2)
          DYJ =DYYC(J)
          DZK =DZZC(K)
          DYZ =DYJ*DZK
          VOL =DYZ*DXI
          CONDW=(DXM1+DXI)*DXI*DXM1*COND(NI+3,J,K)*COND(NI+2,J,K)/
     &         (DXI*COND(NI+3,J,K)+DXM1*COND(NI+2,J,K))
          QCONW=DYZ*CONDW      *(T(NI+2,J,K)-T(NI+3,J,K))*2.0/(DXI+DXM1)
          QCONE=DYZ*COND(2,J,K)*(T(NI+4,J,K)-T(NI+3,J,K))*2.0/DXI
          QRADE=SU(NI+3,J,K)
          T(NI+3,J,K)=TOD(NI+3,J,K)+DTIME*(QCONW+QCONE+QRADE)/
     &               (VOL*CPM(NI+3,J,K))
          T(NI+4,J,K)=(2.0*COND(NI+3,J,K)*T(NI+3,J,K)+HCOEF*TINF*DXI)/
     &               (HCOEF*DXI+2.0*COND(NI+3,J,K))

          QSCONE=QSCONE+QCONW
          QSRADE=QSRADE+QRADE
  80  CONTINUE

C *** CALCULATE THE ENERGY LOST THROUGH (OR CONSUMED BY)
C     1) THE CAVITY  WALLS; 2) CAVITY  AIR; 3) DUCT

C *** TANK  AIR
       WERR=0.
       WAIR=0.
       DO 25 I=3,NI+2
       DO 25 J=3,NJ+2
       DO 25 K=3,NK+2
          IF(NOD(I,J,K).EQ.1) GO TO 25
          DXI=DXXC(I)
          DYJ=DYYC(J)
          DZK=DZZC(K)

C *** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
          DXP1=DXXC(I+1)
          DXI =DXXC(I)
          DXM1=DXXC(I-1)

          DYP1=DYYC(J+1)
          DYJ =DYYC(J)
          DYM1=DYYC(J-1)

          DZP1=DZZC(K+1)
          DZK =DZZC(K)
          DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
          DXN=DXXC(I)
          DXS=DXXC(I)
```

```
          DXF=DXXC(I)
          DXB=DXXC(I)

          DYF=DYYC(J)
          DYB=DYYC(J)
          DYE=DYYC(J)
          DYW=DYYC(J)

          DZE=DZZC(K)
          DZW=DZZC(K)
          DZN=DZZC(K)
          DZS=DZZC(K)

C *** DEFINE AREA OF THE CONTROL VOLUME
          DXYF=DXF*DYF
          DXYB=DXB*DYB
          DYZE=DYE*DZE
          DYZW=DYW*DZW
          DZXN=DZN*DXN
          DZXS=DZS*DXS

          VOL=DXI*DYJ*DZK
          VOLDT=VOL/DTIME

          RN=(R(I,J,K)*DYP1+R(I,J+1,K)*DYJ)/(DYP1+DYJ)
          RS=(R(I,J,K)*DYM1+R(I,J-1,K)*DYJ)/(DYM1+DYJ)
          RE=(R(I,J,K)*DXP1+R(I+1,J,K)*DXI)/(DXP1+DXI)
          RW=(R(I,J,K)*DXM1+R(I-1,J,K)*DXI)/(DXM1+DXI)
          RF=(R(I,J,K)*DZP1+R(I,J,K+1)*DZK)/(DZP1+DZK)
          RB=(R(I,J,K)*DZM1+R(I,J,K-1)*DZK)/(DZM1+DZK)

          CN=RN*V(I,J+1,K)*DZXN
          CS=RS*V(I,J  ,K)*DZXS
          CE=RE*U(I+1,J,K)*DYZE
          CW=RW*U(I  ,J,K)*DYZW
          CF=RF*W(I,J,K+1)*DXYF
          CB=RB*W(I,J,K  )*DXYB

          WERR=WERR+T(I,J,K)*CPM(I,J,K)*SMP(I,J,K)
          WRES=WRES+SMP(I,J,K)
          WAIR=WAIR+(T(I,J,K)*R(I,J,K)-TOD(I,J,K)*ROD(I,J,K))*
     &         CPM(I,J,K)*VOLDT
   25 CONTINUE

C *** SUM UP THE TOTAL WALT AT THAT TIME STEP (DIMENSIONLESS)
       WINS=QQQ/H**3

C *** SUM UP THE TOTAL WALT LOST TO THE WALLS
       QTCON=QSCONF+QSCONB+QSCONS+QSCONN+QSCONW+QSCONE
       QTRAD=QSRADF+QSRADB+QSRADS+QSRADN+QSRADW+QSRADE
       WWAL=QTCON+QTRAD

C *** EQUIVALENT INTERNAL HEAT SOURCE DUE TO RADIATION
       QR=QTRAD*U0*CP0*RHO0*TA*H**2

C *** TOTAL WALT EXHAUSTED THROUGH THE DUCT
```

```
      IF (NMS.EQ.0) THEN
         WWFAN=0.0
      ELSE
         WWFAN=8000.*CPM(12,3,12)*(T(12,3,12)-1.0)*R(12,3,12)/
     &         (60.*H**2*U0)
      ENDIF

C *** THE ENERGY CALCULATION
      QSIN=QSIN+WINS*DTIME
      QSWER=QSWER+WERR*DTIME
      QSWAL=QSWAL+WWAL*DTIME
      QSAIR=QSAIR+WAIR*DTIME
      QSFAN=QSFAN+WWFAN*DTIME

      RETURN
      END

*******************************************************************************
*******************************************************************************
      SUBROUTINE CALU
*******************************************************************************
*CALCULATES THE U COMPONENT OF THE VELOCITY

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL20/SIG11(25,25,15),SIG12(25,25,15),SIG22(25,25,15),
     &            SIG13(25,25,15),SIG23(25,25,15),SIG33(25,25,15)
      COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
     &            JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
      COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
     &            COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
     &            WOD(25,25,15)
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
     &            CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
     &            WPD(25,25,15)
      COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
     &            SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
     &            DV(25,25,15),DW(25,25,15)
      COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
     &            AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
     &            SU(25,25,15),RI(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

C *** CALCULATE COEFFICIENTS
      DO 100 K=2,NK+3
      DO 100 J=2,NJ+3
      DO 100 I=3,NI+3

C *** CENTRAL LENGTH OF THE U CONTROL VOLUME
         DXP1=DXXS(I+1)
```

```
          DXI =DXXS(I)
          DXM1=DXXS(I-1)

          DYP1=DYYC(J+1)
          DYJ =DYYC(J)
          DYM1=DYYC(J-1)

          DZP1=DZZC(K+1)
          DZK =DZZC(K)
          DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
          DXN=DXXS(I)
          DXS=DXXS(I)
          DXF=DXXS(I)
          DXB=DXXS(I)

          DYF=DYYC(J)
          DYB=DYYC(J)
          DYE=DYYC(J)
          DYW=DYYC(J)

          DZE=DZZC(K)
          DZW=DZZC(K)
          DZN=DZZC(K)
          DZS=DZZC(K)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR U
          DXEE=DXXC(I+1)
          DXE =DXXC(I)
          DXW =DXXC(I-1)
          DXWW=DXXC(I-2)

          DYNN=DYYS(J+2)
          DYN =DYYS(J+1)
          DYS =DYYS(J)
          DYSS=DYYS(J-1)

          DZFF=DZZS(K+2)
          DZF =DZZS(K+1)
          DZB =DZZS(K)
          DZBB=DZZS(K-1)

C *** DEFINE THE AREA OF THE CONTROL VOLUME
          DXYF=DXF*DYF
          DXYB=DXB*DYB
          DYZE=DYE*DZE
          DYZW=DYW*DZW
          DZXN=DZN*DXN
          DZXS=DZS*DXS

          VOL=DXI*DYJ*DZK
          VOLDT=VOL/DTIME

          ZXOYN=DZXN/DYN
          ZXOYS=DZXS/DYS
```

```
          XYOZF=DXYF/DZF
          XYOZB=DXYB/DZB
          YZOXE=DYZE/DXE
          YZOXW=DYZW/DXW

C *** USE SINGLE AND BI-LINEAR INTERPOLATION TO EVALUATE
C     PHYSICAL PROPERTIES AND FLUX ON THE SURFACES.
          GNE=SILIN(R(I  ,J+1,K),R(I  ,J,K),DYP1,DYJ)*V(I  ,J+1,K)
          GNW=SILIN(R(I-1,J+1,K),R(I-1,J,K),DYP1,DYJ)*V(I-1,J+1,K)
          GSE=SILIN(R(I  ,J-1,K),R(I  ,J,K),DYM1,DYJ)*V(I  ,J  ,K)
          GSW=SILIN(R(I-1,J-1,K),R(I-1,J,K),DYM1,DYJ)*V(I-1,J  ,K)

          GE =SILIN(R(I+1,J,K),R(I  ,J,K),DXEE,DXE)*U(I+1,J,K)
          GP =SILIN(R(I-1,J,K),R(I  ,J,K),DXW ,DXE)*U(I  ,J,K)
          GW =SILIN(R(I-2,J,K),R(I-1,J,K),DXWW,DXW)*U(I-1,J,K)

          GFE=SILIN(R(I  ,J,K+1),R(I  ,J,K),DZP1,DZK)*W(I  ,J,K+1)
          GFW=SILIN(R(I-1,J,K+1),R(I-1,J,K),DZP1,DZK)*W(I-1,J,K+1)
          GBE=SILIN(R(I  ,J,K-1),R(I  ,J,K),DZM1,DZK)*W(I  ,J,K  )
          GBW=SILIN(R(I-1,J,K-1),R(I-1,J,K),DZM1,DZK)*W(I-1,J,K  )

C *** MASS FLOW RATE
          CE=0.5*(GE+GP)*DYZE
          CW=0.5*(GP+GW)*DYZW

          CN=SILIN(GNE,GNW,DXE,DXW)*DZXN
          CS=SILIN(GSE,GSW,DXE,DXW)*DZXS

          CF=SILIN(GFE,GFW,DXE,DXW)*DXYF
          CB=SILIN(GBE,GBW,DXE,DXW)*DXYB

C *** VISCOSITY
          VISE=VIS(I  ,J,K)
          VISW=VIS(I-1,J,K)

          VISN=(VIS(I,J+1,K)+VIS(I,J,K)+VIS(I-1,J+1,K)+VIS(I-1,J,K))/4.0
          VISS=(VIS(I,J-1,K)+VIS(I,J,K)+VIS(I-1,J-1,K)+VIS(I-1,J,K))/4.0

          VISF=(VIS(I,J,K+1)+VIS(I,J,K)+VIS(I-1,J,K+1)+VIS(I-1,J,K))/4.0
          VISB=(VIS(I,J,K-1)+VIS(I,J,K)+VIS(I-1,J,K-1)+VIS(I-1,J,K))/4.0

          VISN1=ZXOYN*VISN
          VISS1=ZXOYS*VISS
          VISE1=YZOXE*VISE
          VISW1=YZOXW*VISW
          VISF1=XYOZF*VISF
          VISB1=XYOZB*VISB

C *** QUICK SCHEME
          CEP=(ABS(CE)+CE)*DXE/(DXI *16.)
          CEM=(ABS(CE)-CE)*DXE/(DXP1*16.)
          CWP=(ABS(CW)+CW)*DXW/(DXM1*16.)
          CWM=(ABS(CW)-CW)*DXW/(DXI *16.)

          CNP=(ABS(CN)+CN)*DYP1*DYJ/(8.*DYN*(DYN+DYS ))
          CNM=(ABS(CN)-CN)*DYP1*DYJ/(8.*DYN*(DYN+DYNN))
```

```
      CSP=(ABS(CS)+CS)*DYM1*DYJ/(8.*DYS*(DYS+DYSS))
      CSM=(ABS(CS)-CS)*DYM1*DYJ/(8.*DYS*(DYS+DYN ))

      CFP=(ABS(CF)+CF)*DZP1*DZK/(8.*DZF*(DZF+DZB ))
      CFM=(ABS(CF)-CF)*DZP1*DZK/(8.*DZF*(DZF+DZFF))
      CBP=(ABS(CB)+CB)*DZM1*DZK/(8.*DZB*(DZB+DZBB))
      CBM=(ABS(CB)-CB)*DZM1*DZK/(8.*DZB*(DZB+DZF ))

      AE(I,J,K)= -.5*CE    +CWM*DXW /DXE+CEP+CEM*(1.+DXE/DXEE)+VISE1
      AW(I,J,K)=  .5*CW    +CEP*DXE /DXW+CWM+CWP*(1.+DXW/DXWW)+VISW1
      AN(I,J,K)=(-.5*CN*DYJ+CSM*DYS)/DYN+CNP+CNM*(1.+DYN/DYNN)+VISN1
      AS(I,J,K)=( .5*CS*DYJ+CNP*DYN)/DYS+CSM+CSP*(1.+DYS/DYSS)+VISS1
      AF(I,J,K)=(-.5*CF*DZK+CBM*DZB)/DZF+CFP+CFM*(1.+DZF/DZFF)+VISF1
      AB(I,J,K)=( .5*CB*DZK+CFP*DZF)/DZB+CBM+CBP*(1.+DZB/DZBB)+VISB1

C *** BOUNDARY CONSIDERATION
      IF(I.LT.NI+3) THEN
         AEE=-CEM*DXE/DXEE
         AEER=AEE*UPD(I+2,J,K)
      ELSE
         AEE=0.
         AEER=0.
      ENDIF

      IF (I.GT.3) THEN
         AWW=-CWP*DXW/DXWW
         AWWR=AWW*UPD(I-2,J,K)
      ELSE
         AWW=0.
         AWWR=0.
      ENDIF

      IF (J.LT.NJ+3) THEN
         ANN=-CNM*DYN/DYNN
         ANNR=ANN*UPD(I,J+2,K)
      ELSE
         ANN=0.
         ANNR=0.
      ENDIF

      IF (J.GT.2) THEN
         ASS=-CSP*DYS/DYSS
         ASSR=ASS*UPD(I,J-2,K)
      ELSE
         ASS=0.
         ASSR=0.
      ENDIF

      IF (K.LT.NK+3) THEN
         AFF=-CFM*DZF/DZFF
         AFFR=AFF*UPD(I,J,K+2)
      ELSE
         AFF=0.
         AFFR=0.
      ENDIF
```

```
          IF (K.GT.2) THEN
             ABB=-CBP*DZB/DZBB
             ABBR=ABB*UPD(I,J,K-2)
          ELSE
             ABB=0.
             ABBR=0.
          ENDIF

C *** MODIFICATION FOR DECK BOUNDARIES
          IF (NOD(I-2,J,K).NE.0) THEN
             AWW=0.0
             AWWR=0.0
          ENDIF

          IF (NOD(I+1,J,K).NE.0) THEN
             AEE=0.0
             AEER=0.0
          ENDIF

          IF (NOD(I,J-1,K).NE.0) THEN
             ASS=0.0
             ASSR=0.0
          ENDIF

          IF (NOD(I,J+1,K).NE.0) THEN
             ANN=0.0
             ANNR=0.0
          ENDIF

          IF (NOD(I,J,K-1).NE.0) THEN
             ABB=0.0
             ABBR=0.0
          ENDIF

          IF (NOD(I,J,K+1).NE.0) THEN
             AFF=0.0
             AFFR=0.0
          ENDIF

C *** SU FROM NORMAL STRESS
          RE=(SIG11(I  ,J,K)-(U(I+1,J,K)-U(I  ,J,K))*VISE/DXE)*DYZE
          RW=(SIG11(I-1,J,K)-(U(I  ,J,K)-U(I-1,J,K))*VISW/DXW)*DYZW

          RN=(SIG12(I,J+1,K)-(U(I,J+1,K)-U(I,J  ,K))*VISN/DYN)*DZXN
          RS=(SIG12(I,J  ,K)-(U(I,J  ,K)-U(I,J-1,K))*VISS/DYS)*DZXS

          RF=(SIG13(I,J,K+1)-(U(I,J,K+1)-U(I,J,K  ))*VISF/DZF)*DXYF
          RB=(SIG13(I,J,K  )-(U(I,J,K  )-U(I,J,K-1))*VISB/DZB)*DXYB

C *** SU FROM CURVED STRESSES AND ACCELERATIONS
          AVG12=0.5*(SIG12(I,J+1,K  )+SIG12(I,J,K))
          AVG13=0.5*(SIG13(I,J  ,K+1)+SIG13(I,J,K))

          AVG22=SILIN(SIG22(I,J,K),SIG22(I-1,J,K),DXE,DXW)
          AVG33=SILIN(SIG33(I,J,K),SIG33(I-1,J,K),DXE,DXW)
```

```
          AU1=U(I,J,K)
          AU2=BILIN(V(I  ,J+1,K),V(I  ,J,K),DYJ,DYJ,
     &              V(I-1,J+1,K),V(I-1,J,K),DYJ,DYJ,DXE,DXW)
          AU3=BILIN(W(I  ,J,K+1),W(I  ,J,K),DZK,DZK,
     &              W(I-1,J,K+1),W(I-1,J,K),DZK,DZK,DXE,DXW)

          AR=SILIN(R(I,J,K),R(I-1,J,K),DXE,DXW)

          ARU12=AR*AU1*AU2
          ARU13=AR*AU1*AU3
          ARU22=AR*AU2*AU2
          ARU33=AR*AU3*AU3

          RRY=(AVG12-ARU12)*DZK*(DXN-DXS)
          RRZ=(AVG13-ARU13)*DYJ*(DXF-DXB)
          RRX=(AVG22-ARU22)*DZK*(DYE-DYW)+(AVG33-ARU33)*DYJ*(DZE-DZW)

          AP(I,J,K)=AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)+AF(I,J,K)+
     &              AB(I,J,K)+AEE+AWW+ANN+ASS+AFF+ABB
          SP(I,J,K)=-(ROD(I,J,K)*DXW+ROD(I-1,J,K)*DXE)*VOLDT/(DXW+DXE)
          SU(I,J,K)=-SP(I,J,K)*UOD(I,J,K)+DYJ*DZK*(P(I-1,J,K)-P(I,J,K))+
     &              AEER+AWWR+ANNR+ASSR+AFFR+ABBR+RE-RW+RN-RS+RF-RB+RRY+
     &              RRZ-RRX
  100 CONTINUE

C *** TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU

C *** Y DIRECTION
      DO 500 K=2,NK+3
      DO 500 I=3,NI+3
         SP(I,2   ,K)=SP(I,2   ,K)-AS(I,2   ,K)
         SP(I,NJ+3,K)=SP(I,NJ+3,K)-AN(I,NJ+3,K)
         AN(I,NJ+3,K)=0.
         AS(I,2   ,K)=0.
  500 CONTINUE

C *** X DIRECTION
      DO 502 K=2,NK+3
      DO 502 J=2,NJ+3
         AW(3   ,J,K)=0.0
         AE(NI+3,J,K)=0.0
  502 CONTINUE

C *** Z DIRECTION
      DO 600 I=3,NI+3
      DO 600 J=2,NJ+3
         SP(I,J,2   )=SP(I,J,2   )-AB(I,J,2   )
         SP(I,J,NK+3)=SP(I,J,NK+3)-AF(I,J,NK+3)
         AF(I,J,NK+3)=0.
         AB(I,J,2   )=0.
 600  CONTINUE

C *** MODIFICATION FOR DECK BOUNDARIES
      IF (NCHIP.EQ.0) GOTO 201
      DO 101 N=1,NCHIP
         IB  =ICHPB(N)
```

```
            IE  =IB+NCHPI(N)-1
            JB  =JCHPB(N)
            JE  =JB+NCHPJ(N)-1
            KB  =KCHPB(N)
            KE  =KB+NCHPK(N)-1

            DO 102 J=JB,JE-1
            DO 102 K=KB,KE-1
               AE(IB-1,J,K)=0.0
               AW(IE+1,J,K)=0.0
  102       CONTINUE

            DO 103 I=IB,IE
            DO 103 K=KB,KE-1
               SP(I,JB-1,K)=SP(I,JB-1,K)-AN(I,JB-1,K)
               AN(I,JB-1,K)=0.0
               SP(I,JE  ,K)=SP(I,JE  ,K)-AS(I,JE  ,K)
               AS(I,JE  ,K)=0.0
  103       CONTINUE

            DO 106 I=IB,IE
            DO 106 J=JB,JE-1
               SP(I,J,KB-1)=SP(I,J,KB-1)-AF(I,J,KB-1)
               AF(I,J,KB-1)=0.0
               SP(I,J,KE  )=SP(I,J,KE  )-AB(I,J,KE  )
               AB(I,J,KE  )=0.0
  106       CONTINUE

C *** FOR THE CELLS INSIDE OF THE DECKS
            DO 104 I=IB,IE
            DO 104 J=JB,JE-1
            DO 104 K=KB,KE-1
               SP(I,J,K)=-1.0E2
               AW(I,J,K)=0.
               AE(I,J,K)=0.
               AS(I,J,K)=0.
               AN(I,J,K)=0.
               AB(I,J,K)=0.
               AF(I,J,K)=0.
               SU(I,J,K)=0.
  104       CONTINUE
  101 CONTINUE

C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
  201 DO 301 K=2,NK+3
      DO 301 J=2,NJ+3
      DO 301 I=3,NI+3
         DYJ=DYYC(J)
         DZK=DZZC(K)
         DYZ=DYJ*DZK
         AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
         DU(I,J,K)=DYZ/AP(I,J,K)
  301 CONTINUE

C *** SOLVE FOR U
      CALL TRID (4,3,3,NI+2,NJ+2,NK+2,U)
```

```
C *** RESET THE VELOCITY INSIDE OF DECK
      IF (NCHIP.EQ.0) GOTO 111
      DO 110 N=1,NCHIP
         IB=ICHPB(N)
         IE=IB+NCHPI(N)-1
         JB=JCHPB(N)
         JE=JB+NCHPJ(N)-1
         KB=KCHPB(N)
         KE=KB+NCHPK(N)-1
         DO 108 I=IB,IE
         DO 108 J=JB,JE-1
         DO 108 K=KB,KE-1
            U(I,J,K)=0.0
  108    CONTINUE
  110 CONTINUE

  111 RETURN
      END

*******************************************************************************
*******************************************************************************
      SUBROUTINE CALV
*******************************************************************************
*CALCULATES THE V COMPONENT OF THE VELOCITY

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL20/SIG11(25,25,15),SIG12(25,25,15),SIG22(25,25,15),
     &            SIG13(25,25,15),SIG23(25,25,15),SIG33(25,25,15)
      COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
     &            JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
      COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
     &            COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
     &            WOD(25,25,15)
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
     &            CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
     &            WPD(25,25,15)
      COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
     &            SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
     &            DV(25,25,15),DW(25,25,15)
      COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
     &            AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
     &            SU(25,25,15),RI(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

C *** CALCULATE COEFFICIENTS
      DO 100 K=2,NK+3
      DO 100 J=3,NJ+3
      DO 100 I=2,NI+3
```

```
C *** CENTRAL LENGTH OF THE V CONTROL VOLUME
        DXP1=DXXC(I+1)
        DXI =DXXC(I)
        DXM1=DXXC(I-1)

        DYP1=DYYS(J+1)
        DYJ =DYYS(J)
        DYM1=DYYS(J-1)

        DZP1=DZZC(K+1)
        DZK =DZZC(K)
        DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
        DXN=DXXC(I)
        DXS=DXXC(I)
        DXF=DXXC(I)
        DXB=DXXC(I)

        DYF=DYYS(J)
        DYB=DYYS(J)
        DYE=DYYS(J)
        DYW=DYYS(J)

        DZE=DZZC(K)
        DZW=DZZC(K)
        DZN=DZZC(K)
        DZS=DZZC(K)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR V
        DXEE=DXXS(I+2)
        DXE =DXXS(I+1)
        DXW =DXXS(I)
        DXWW=DXXS(I-1)

        DYNN=DYYC(J+1)
        DYN =DYYC(J)
        DYS =DYYC(J-1)
        DYSS=DYYC(J-2)

        DZFF=DZZC(K+2)
        DZF =DZZC(K+1)
        DZB =DZZC(K)
        DZBB=DZZC(K-1)

C *** DEFINE THE AREA OF THE CONTROL VOLUME
        DXYF=DXF*DYF
        DXYB=DXB*DYB
        DYZE=DYE*DZE
        DYZW=DYW*DZW
        DZXN=DZN*DXN
        DZXS=DZS*DXS

        VOL  =DXI*DYJ*DZK
        VOLDT=VOL/DTIME
```

```
          ZXOYN=DZXN/DYN
          ZXOYS=DZXS/DYS
          XYOZF=DXYF/DZF
          XYOZB=DXYB/DZB
          YZOXE=DYZE/DXE
          YZOXW=DYZW/DXW

C *** USE SINGLE AND BI-LINEAR INTERPOLATION TO EVALUATE
C     PHYSICAL PROPERTIES AND FLUX ON THE SURFACES.
          GEN=SILIN(R(I+1,J  ,K),R(I,J  ,K),DXP1,DXI)*U(I+1,J  ,K)
          GES=SILIN(R(I+1,J-1,K),R(I,J-1,K),DXP1,DXI)*U(I+1,J-1,K)
          GWN=SILIN(R(I-1,J  ,K),R(I,J  ,K),DXM1,DXI)*U(I  ,J  ,K)
          GWS=SILIN(R(I-1,J-1,K),R(I,J-1,K),DXM1,DXI)*U(I  ,J-1,K)

          GN =SILIN(R(I,J+1,K),R(I,J  ,K),DYNN,DYN)*V(1,J+1,K)
          GP =SILIN(R(I,J-1,K),R(I,J  ,K),DYS ,DYN)*V(I,J  ,K)
          GS =SILIN(R(I,J-2,K),R(I,J-1,K),DYSS,DYS)*V(I,J-1,K)

          GFN=SILIN(R(I,J  ,K+1),R(I,J  ,K),DZP1,DZK)*W(I,J  ,K+1)
          GFS=SILIN(R(I,J-1,K+1),R(I,J-1,K),DZP1,DZK)*W(I,J-1,K+1)
          GBN=SILIN(R(I,J  ,K-1),R(I,J  ,K),DZM1,DZK)*W(I,J  ,K  )
          GBS=SILIN(R(I,J-1,K-1),R(I,J-1,K),DZM1,DZK)*W(I,J-1,K  )

C *** MASS FLOW RATE
          CN=0.5*(GN+GP)*DZXN
          CS=0.5*(GP+GS)*DZXS
          CE=SILIN(GEN,GES,DYN,DYS)*DYZE
          CW=SILIN(GWN,GWS,DYN,DYS)*DYZW
          CF=SILIN(GFN,GFS,DYN,DYS)*DXYF
          CB=SILIN(GBN,GBS,DYN,DYS)*DXYB

C *** VISCOSITY
          VISN= VIS(I,J  ,K)
          VISS= VIS(I,J-1,K)
          VISE=(VIS(I+1,J,K)+VIS(I,J,K)+VIS(I+1,J-1,K)+VIS(I,J-1,K))/4.0
          VISW=(VIS(I-1,J,K)+VIS(I,J,K)+VIS(I-1,J-1,K)+VIS(I,J-1,K))/4.0
          VISF=(VIS(I,J,K+1)+VIS(I,J,K)+VIS(I,J-1,K+1)+VIS(I,J-1,K))/4.0
          VISB=(VIS(I,J,K-1)+VIS(I,J,K)+VIS(I,J-1,K-1)+VIS(I,J-1,K))/4.0

          VISN1=ZXOYN*VISN
          VISS1=ZXOYS*VISS
          VISE1=YZOXE*VISE
          VISW1=YZOXW*VISW
          VISF1=XYOZF*VISF
          VISB1=XYOZB*VISB

C *** QUICK SCHEME
          CEP=(ABS(CE)+CE)*DXP1*DXI/(DXE*(DXE+DXW )*8.)
          CEM=(ABS(CE)-CE)*DXP1*DXI/(DXE*(DXE+DXEE)*8.)
          CWP=(ABS(CW)+CW)*DXM1*DXI/(DXW*(DXW+DXWW)*8.)
          CWM=(ABS(CW)-CW)*DXM1*DXI/(DXW*(DXW+DXE )*8.)

          CNP=(ABS(CN)+CN)*DYN/(DYJ *16.)
          CNM=(ABS(CN)-CN)*DYN/(DYP1*16.)
          CSP=(ABS(CS)+CS)*DYS/(DYM1*16.)
```

```
      CSM=(ABS(CS)-CS)*DYS/(DYJ *16.)

      CFP=(ABS(CF)+CF)*DZP1*DZK/(DZF*(DZF+DZB )*8.)
      CFM=(ABS(CF)-CF)*DZP1*DZK/(DZF*(DZF+DZFF)*8.)
      CBP=(ABS(CB)+CB)*DZM1*DZK/(DZB*(DZB+DZBB)*8.)
      CBM=(ABS(CB)-CB)*DZM1*DZK/(DZB*(DZB+DZF )*8.)

      AE(I,J,K)=(-.5*CE*DXI+CWM*DXW)/DXE+CEP+CEM*(1.+DXE/DXEE)+VISE1
      AW(I,J,K)=( .5*CW*DXI+CEP*DXE)/DXW+CWM+CWP*(1.+DXW/DXWW)+VISW1
      AN(I,J,K)= -.5*CN    +CSM*DYS /DYN+CNP+CNM*(1.+DYN/DYNN)+VISN1
      AS(I,J,K)=  .5*CS    +CNP*DYN /DYS+CSM+CSP*(1.+DYS/DYSS)+VISS1
      AF(I,J,K)=(-.5*CF*DZK+CBM*DZB)/DZF+CFP+CFM*(1.+DZF/DZFF)+VISF1
      AB(I,J,K)=( .5*CB*DZK+CFP*DZF)/DZB+CBM+CBP*(1.+DZB/DZBB)+VISB1

C *** BOUNDARY CONSIDERATION
      IF (I.LT.NI+3) THEN
         AEE=-CEM*DXE/DXEE
         AEER=AEE*VPD(I+2,J,K)
      ELSE
         AEE=0.
         AEER=0.
      ENDIF

      IF (I.GT.2) THEN
         AWW=-CWP*DXW/DXWW
         AWWR=AWW*VPD(I-2,J,K)
      ELSE
         AWW=0.
         AWWR=0.
      ENDIF

      IF (J.LT.NJ+3) THEN
         ANN=-CNM*DYN/DYNN
         ANNR=ANN*VPD(I,J+2,K)
      ELSE
         ANN=0.
         ANNR=0.
      ENDIF

      IF (J.GT.3) THEN
         ASS=-CSP*DYS/DYSS
         ASSR=ASS*VPD(I,J-2,K)
      ELSE
         ASS=0.
         ASSR=0.
      ENDIF

      IF (K.LT.NK+3) THEN
         AFF=-CFM*DZF/DZFF
         AFFR=AFF*VPD(I,J,K+2)
      ELSE
         AFF=0.
         AFFR=0.
      ENDIF

      IF (K.GT.2) THEN
```

```
            ABB=-CBP*DZB/DZBB
            ABBR=ABB*VPD(I,J,K-2)
         ELSE
            ABB=0.
            ABBR=0.
         ENDIF

C *** MODIFICATION FOR DECK BOUNDARIES
         IF (NOD(I-1,J,K).NE.0) THEN
            AWW=0.0
            AWWR=0.0
         ENDIF

         IF (NOD(I+1,J,K).NE.0) THEN
            AEE=0.0
            AEER=0.0
         ENDIF

         IF (NOD(I,J-2,K).NE.0) THEN
            ASS=0.0
            ASSR=0.0
         ENDIF

         IF (NOD(I,J+1,K).NE.0) THEN
            ANN=0.0
            ANNR=0.0
         ENDIF

         IF (NOD(I,J,K-1).NE.0) THEN
            ABB=0.0
            ABBR=0.0
         ENDIF

         IF (NOD(I,J,K+1).EQ.0) THEN
            AFF=0.0
            AFFR=0.0
         ENDIF

C *** SU FROM NORMAL STRESS
         RN=(SIG22(I,J  ,K)-(V(I,J+1,K)-V(I,J  ,K))*VISN/DYN)*DZXN
         RS=(SIG22(I,J-1,K)-(V(I,J  ,K)-V(I,J-1,K))*VISS/DYS)*DZXS
         RE=(SIG12(I+1,J,K)-(V(I+1,J,K)-V(I  ,J,K))*VISE/DXE)*DYZE
         RW=(SIG12(I  ,J,K)-(V(I  ,J,K)-V(I-1,J,K))*VISW/DXW)*DYZW
         RF=(SIG23(I,J,K+1)-(V(I,J,K+1)-V(I,J,K  ))*VISF/DZF)*DXYF
         RB=(SIG23(I,J,K  )-(V(I,J,K  )-V(I,J,K-1))*VISB/DZB)*DXYB

C *** SU FROM CURVED STRESSES AND ACCELERATIONS
         AVG12= 0.5*(SIG12(I+1,J,K  )+SIG12(I,J  ,K))
         AVG23= 0.5*(SIG23(I  ,J,K+1)+SIG23(I,J  ,K))
         AVG11=SILIN(SIG11(I  ,J,K  ),SIG11(I,J-1,K),DYN,DYS)
         AVG33=SILIN(SIG33(I  ,J,K  ),SIG33(I,J-1,K),DYN,DYS)

         AU2=V(I,J,K)
         AU1=BILIN(U(I+1,J  ,K  ),U(I,J  ,K),DXI,DXI,
     &             U(I+1,J-1,K  ),U(I,J-1,K),DXI,DXI,DYN,DYS)
         AU3=BILIN(W(I  ,J  ,K+1),W(I,J  ,K),DZK,DZK,
```

```
     &             W(I   ,J-1,K+1),W(I,J-1,K),DZK,DZK,DYN,DYS)

          AR=SILIN(R(I,J,K),R(I,J-1,K),DYN,DYS)

          ARU12=AR*AU1*AU2
          ARU23=AR*AU2*AU3
          ARU11=AR*AU1*AU1
          ARU33=AR*AU3*AU3

          RRX=(AVG12-ARU12)*DZK*(DYE-DYW)
          RRZ=(AVG23-ARU23)*DXI*(DYF-DYB)
          RRY=(AVG11-ARU11)*DZK*(DXN-DXS)+(AVG33-ARU33)*DXI*(DZN-DZS)

          AP(I,J,K)=AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)+
     &             AF(I,J,K)+AB(I,J,K)+AEE+AWW+ANN+ASS+AFF+ABB
          SP(I,J,K)=-(ROD(I,J,K)*DYS+ROD(I,J-1,K)*DYN)*VOLDT/(DYS+DYN)
          SU(I,J,K)=-SP(I,J,K)*VOD(I,J,K)+DZK*DXI*(P(I,J-1,K)-P(I,J,K))+
     &             AEER+AWWR+ANNR+ASSR+AFFR+ABBR+RE-RW+RN-RS+RF-RB+RRX+
     &             RRZ-RRY
  100 CONTINUE

C *** TAKE CARE OF B.C. THRU AN,AS,AE,AW,AF,AB,SP AND SU

C *** Y DIRECTION
      DO 500 K=2,NK+3
      DO 500 I=2,NI+3
         AS(I,3   ,K)=0.
         AN(I,NJ+3,K)=0.
  500 CONTINUE

C *** X DIRECTION
      DO 502 K=2,NK+3
      DO 502 J=3,NJ+3
         SP(2   ,J,K)=SP(2   ,J,K)-AW(2   ,J,K)
         SP(NI+3,J,K)=SP(NI+3,J,K)-AE(NI+3,J,K)
         AW(2   ,J,K)=0.0
         AE(NI+3,J,K)=0.0
  502 CONTINUE

C *** Z DIRECTION
      DO 600 I=2,NI+3
      DO 600 J=3,NJ+3
         SP(I,J,2   )=SP(I,J,2   )-AB(I,J,2   )
         SP(I,J,NK+3)=SP(I,J,NK+3)-AF(I,J,NK+3)
         AF(I,J,NK+3)=0.
         AB(I,J,2   )=0.
  600 CONTINUE

C *** MODIFICATION FOR DECK BOUNDARIES
      IF (NCHIP.EQ.0) GOTO 201
      DO 101 N=1,NCHIP
         IB  =ICHPB(N)
         IE  =IB+NCHPI(N)-1
         JB  =JCHPB(N)
         JE  =JB+NCHPJ(N)-1
         KB  =KCHPB(N)
```

```
            KE  =KB+NCHPK(N)-1
            DO 102 J=JB,JE
            DO 102 K=KB,KE-1
               SP(IB-1,J,K)=SP(IB-1,J,K)-AE(IB-1,J,K)
               AE(IB-1,J,K)=0.0
               SP(IE  ,J,K)=SP(IE  ,J,K)-AW(IE  ,J,K)
               AW(IE  ,J,K)=0.0
  102       CONTINUE

            DO 103 I=IB,IE-1
            DO 103 K=KB,KE-1
               AN(I,JB-1,K)=0.0
               AS(I,JE+1,K)=0.0
  103       CONTINUE

            DO 106 I=IB,IE-1
            DO 106 J=JB,JE
               SP(I,J,KB-1)=SP(I,J,KB-1)-AF(I,J,KB-1)
               AF(I,J,KB-1)=0.0
               SP(I,J,KE  )=SP(I,J,KE  )-AB(I,J,KE  )
               AB(I,J,KE  )=0.0
  106       CONTINUE

C *** MODIFICATION FOR THE CELLS INSIDE OF THE DECKS
            DO 104 I=IB,IE-1
            DO 104 J=JB,JE
            DO 104 K=KB,KE-1
               SP(I,J,K)=-1.0E2
               AW(I,J,K)=0.
               AE(I,J,K)=0.
               AS(I,J,K)=0.
               AN(I,J,K)=0.
               AB(I,J,K)=0.
               AF(I,J,K)=0.
               SU(I,J,K)=0.
  104       CONTINUE
  101 CONTINUE

C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
  201 DO 300 K=2,NK+3
      DO 300 J=3,NJ+3
      DO 300 I=2,NI+3
         DXI=DXXC(I)
         DZK=DZZC(K)
         DZX=DZK*DXI
         AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
         DV(I,J,K)=DZX/AP(I,J,K)
  300 CONTINUE

C *** SOLVE FOR V
      CALL TRID (3,4,3,NI+2,NJ+2,NK+2,V)

C *** RESET THE VELOCITY INSIDE OF THE DECKS
      IF (NCHIP.EQ.0) GOTO 111
      DO 110 N=1,NCHIP
         IB=ICHPB(N)
```

```
         IE=IB+NCHPI(N)-1
         JB=JCHPB(N)
         JE=JB+NCHPJ(N)-1
         KB=KCHPB(N)
         KE=KB+NCHPK(N)-1
         DO 108 I=IB,IE-1
         DO 108 J=JB,JE
         DO 108 K=KB,KE-1
            V(I,J,K)=0.0
  108    CONTINUE
  110 CONTINUE

  111 RETURN
      END

*************************************************************************
*************************************************************************
      SUBROUTINE CALVIS
*************************************************************************
*     THIS SUBROUTINE CALCULATES THE TURBULENT VISCOSITY AND UPDATES   *
*     THE VISCOSITY MATRIX                                             *
*************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL14/HCOEF,CNT,ABTURB,BTURB,VISL,VISMAX
      COMMON/BL16/U0,UGRT,BUOY,CP0,PRT,COND0,VIS0,RHO0,
     &            TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
     &            SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
     &            DV(25,25,15),DW(25,25,15)
      COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
     &            AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
     &            SU(25,25,15),RI(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

C *** CALCULATE LOCAL SHEAR AND VISCOSITY VIS(I,J,K)

C *** SPECIFY LOCAL TURBULENT LENGTH SCALES SMPP(I,J,K)
      DO 611 K=3,NK+2
      DO 611 J=3,NJ+2
      DO 611 I=3,NI+2

C *** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
         DXP1=DXXC(I+1)
         DXI =DXXC(I)
         DXM1=DXXC(I-1)

         DYP1=DYYC(J+1)
         DYJ =DYYC(J)
         DYM1=DYYC(J-1)
```

```
      DZP1=DZZC(K+1)
      DZK =DZZC(K)
      DZM1=DZZC(K-1)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR T
      DXE =DXXS(I+1)
      DXW =DXXS(I)

      DYN =DYYS(J+1)
      DYS =DYYS(J)

      DZF =DZZS(K+1)
      DZB =DZZS(K)

C *** CACULATE DV/DX,D2V/DX2,DU/DX,D2U/DX2,DW/DX AND D2W/DX2
      DUDX  =    (U(I+1,J,K)-U(I  ,J,K))/DXI
      DUDXW =0.5*(U(I+1,J,K)-U(I-1,J,K))/DXW
      DUDXE =0.5*(U(I+2,J,K)-U(I  ,J,K))/DXE
      D2UDX2=(DUDXE-DUDXW)/DXI

      DVDXW =0.5*(V(I,J+1,K)+V(I,J,K)-V(I-1,J+1,K)-V(I-1,J,K))/DXW
      DVDXE =0.5*(V(I+1,J+1,K)+V(I+1,J,K)-V(I,J+1,K)-V(I,J,K))/DXE
      DVDX  =0.5*(DVDXE+DVDXW)
      D2VDX2=    (DVDXE-DVDXW)/DXI

      DWDXW =0.5*(W(I,J,K+1)+W(I,J,K)-W(I-1,J,K+1)-W(I-1,J,K))/DXW
      DWDXE =0.5*(W(I+1,J,K+1)+W(I+1,J,K)-W(I,J,K+1)-W(I,J,K))/DXE
      DWDX  =0.5*(DWDXE+DWDXW)
      D2WDX2=    (DWDXE-DWDXW)/DXI

C *** CALCULATE DU/DY,D2U/DY2,DV/DY,D2V/DY2,DW/DY AND D2W/DY2
      DVDY  =    (V(I,J+1,K)-V(I,J  ,K))/DYJ
      DVDYS =0.5*(V(I,J+1,K)-V(I,J-1,K))/DYS
      DVDYN =0.5*(V(I,J+2,K)-V(I,J  ,K))/DYN
      D2VDY2=(DVDYN-DVDYS)/DYJ

      DUDYS =0.5*(U(I+1,J,K)+U(I,J,K)-U(I+1,J-1,K)-U(I,J-1,K))/DYS
      DUDYN =0.5*(U(I+1,J+1,K)+U(I,J+1,K)-U(I+1,J,K)-U(I,J,K))/DYN
      DUDY  =0.5*(DUDYN+DUDYS)
      D2UDY2=    (DUDYN-DUDYS)/DYJ

      DWDYS =0.5*(W(I,J,K+1)+W(I,J,K)-W(I,J-1,K+1)-W(I,J-1,K))/DYS
      DWDYN =0.5*(W(I,J+1,K+1)+W(I,J+1,K)-W(I,J,K+1)-W(I,J,K))/DYN
      DWDY  =0.5*(DWDYN+DWDYS)
      D2WDY2=    (DWDYN-DWDYS)/DYJ

C *** CALCULATE DU/DZ,D2U/DZ2,DV/DZ,D2V/DZ2,DW/DZ AND D2W/DZ2
      DWDZ  =    (W(I,J,K+1)-W(I,J,K  ))/DZK
      DWDZF =0.5*(W(I,J,K+2)-W(I,J,K  ))/DZF
      DWDZB =0.5*(W(I,J,K+1)-W(I,J,K-1))/DZB
      D2WDZ2=(DWDZF-DWDZB)/DZK

      DVDZB =0.5*(V(I,J+1,K)+V(I,J,K)-V(I,J+1,K-1)-V(I,J,K-1))/DZB
      DVDZF =0.5*(V(I,J+1,K+1)+V(I,J,K+1)-V(I,J+1,K)-V(I,J,K))/DZF
      DVDZ  =0.5*(DVDZF+DVDZB)
```

```
          D2VDZ2=    (DVDZF-DVDZB)/DZK

          DUDZB =0.5*(U(I+1,J,K)+U(I,J,K)-U(I+1,J,K-1)-U(I,J,K-1))/DZB
          DUDZF =0.5*(U(I+1,J,K+1)+U(I,J,K+1)-U(I+1,J,K)-U(I,J,K))/DZF
          DUDZ  =0.5*(DUDZF+DUDZB)
          D2UDZ2=    (DUDZF-DUDZB)/DZK

C *** CALCULATE THE DENSITY GRADIENT WITH RESPECT TO THE VERTICAL
          DRDGA=(R(I,J,K+1)-REQ(I,J,K+1)-R(I,J,K-1)+REQ(I,J,K-1))/
     &          (DZF+DZB)

C *** CALCULATE STRAIN
          STRAIN=DUDY**2+DVDX**2+DWDX**2+DVDZ**2+DWDY**2+DUDZ**2
          DDO2  =SQRT(STRAIN+DUDX**2+DVDY**2+DWDZ**2)

          IF(DDO2.EQ.0..OR.STRAIN.EQ.0.) THEN
             VIS(I,J,K)=VISL
          ELSE

C *** CALCULATE TURBULENT LENGTH SCALE SMPP(I,J)
             SMP123=SQRT(((U(I+1,J,K)+U(I,J,K))/2.)**2+
     &                   ((V(I,J+1,K)+V(I,J,K))/2.)**2+
     &                   ((W(I,J,K+1)+W(I,J,K))/2.)**2)/DDO2
             SMPP12=DDO2/SQRT(D2UDX2**2+D2UDY2**2+D2UDZ2**2+D2VDX2**2+
     &              D2VDY2**2+D2VDZ2**2+D2WDZ2**2+D2WDX2**2+D2WDY2**2)
             SMPP(I,J,K)=CNT*(SMP123+SMPP12)/2.

C *** CALCULATE RICHARDSON NUMBER
             RI(I,J,K)=-BUOY*DRDGA/(R(I,J,K)*STRAIN)
             ABRIPR=ABTURB+RI(I,J,K)/PRT

             IF(ABRIPR.LT.0.) THEN
                VIS(I,J,K)=VISL
             ELSEIF(ABRIPR.EQ.0.) THEN
                VIS(I,J,K)=VISMAX
             ELSE
                VIS(I,J,K)=VISL+R(I,J,K)*SMPP(I,J,K)**2*
     &                     SQRT(STRAIN)/(BTURB*ABRIPR)
                IF(VIS(I,J,K).GT.VISMAX) VIS(I,J,K)=VISMAX
             ENDIF
          ENDIF
  611 CONTINUE

C *** SPECIFY THE VISCOCITY ON THE BOUNDARY POINT
       DO 110 I=1,NI+4
       DO 110 J=1,NJ+4
          VIS(I,J,NK+3)=VIS(I,J,NK+2)
          VIS(I,J,2   )=VIS(I,J,3   )
  110 CONTINUE

       DO 120 J=1,NJ+4
       DO 120 K=1,NK+4
          VIS(NI+3,J,K)=VIS(NI+2,J,K)
          VIS(2   ,J,K)=VIS(3   ,J,K)
  120 CONTINUE
```

```
      DO 130 K=1,NK+4
      DO 130 I=1,NI+4
         VIS(I,NJ+3,K)=VIS(I,NJ+2,K)
         VIS(I,2   ,K)=VIS(I,3   ,K)
  130 CONTINUE

C *** CALCULATE TURBULENT CONDUCTIVITY
      DO 140 I=1,NI+4
      DO 140 J=1,NJ+4
      DO 140 K=1,NK+4
         IF (NOD(I,J,K).NE.1) COND(I,J,K)=VIS(I,J,K)/PRT
  140 CONTINUE
      RETURN
      END

*******************************************************************************
*******************************************************************************
      SUBROUTINE CALW
*******************************************************************************
*CALCULATES THE W COMPONENT OF THE VELOCITY

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL16/U0,UGRT,BUOY,CP0,PRT,COND0,VIS0,RHO0,
     &            TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
      COMMON/BL20/SIG11(25,25,15),SIG12(25,25,15),SIG22(25,25,15),
     &            SIG13(25,25,15),SIG23(25,25,15),SIG33(25,25,15)
      COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
     &            JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
      COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
     &            COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
     &            WOD(25,25,15)
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
     &            CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
     &            WPD(25,25,15)
      COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
     &            SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
     &            DV(25,25,15),DW(25,25,15)
      COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
     &            AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
     &            SU(25,25,15),RI(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

C *** CALCULATE COEFFICIENTS
      DO 100 K=3,NK+3
      DO 100 J=2,NJ+3
      DO 100 I=2,NI+3

C *** CENTRAL LENGTH OF THE W CONTROL VOLUME
         DXP1=DXXC(I+1)
```

```
            DXI =DXXC(I)
            DXM1=DXXC(I-1)

            DYP1=DYYC(J+1)
            DYJ =DYYC(J)
            DYM1=DYYC(J-1)

            DZP1=DZZS(K+1)
            DZK =DZZS(K)
            DZM1=DZZS(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
            DXN=DXXC(I)
            DXS=DXXC(I)
            DXF=DXXC(I)
            DXB=DXXC(I)

            DYF=DYYC(J)
            DYB=DYYC(J)
            DYE=DYYC(J)
            DYW=DYYC(J)

            DZE=DZZS(K)
            DZW=DZZS(K)
            DZN=DZZS(K)
            DZS=DZZS(K)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME
            DXEE=DXXS(I+2)
            DXE =DXXS(I+1)
            DXW =DXXS(I)
            DXWW=DXXS(I-1)

            DYNN=DYYS(J+2)
            DYN =DYYS(J+1)
            DYS =DYYS(J)
            DYSS=DYYS(J-1)

            DZFF=DZZC(K+1)
            DZF =DZZC(K)
            DZB =DZZC(K-1)
            DZBB=DZZC(K-2)

C *** DEFINE THE AREA OF THE CONTROL VOLUME
            DXYF=DXF*DYF
            DXYB=DXB*DYB
            DYZE=DYE*DZE
            DYZW=DYW*DZW
            DZXN=DZN*DXN
            DZXS=DZS*DXS

            VOL=DXI*DYJ*DZK
            VOLDT=VOL/DTIME

            ZXOYN=DZXN/DYN
            ZXOYS=DZXS/DYS
```

```
          XYOZF=DXYF/DZF
          XYOZB=DXYB/DZB
          YZOXE=DYZE/DXE
          YZOXW=DYZW/DXW

C *** USE SINGLE AND BI-LINEAR INTERPOLATION TO EVALUATE
C     PHYSICAL PROPERTIES AND FLUX ON THE SURFACES.
          GNF=SILIN(R(I,J+1,K  ),R(I,J,K  ),DYP1,DYJ)*V(I,J+1,K  )
          GNB=SILIN(R(I,J+1,K-1),R(I,J,K-1),DYP1,DYJ)*V(I,J+1,K-1)
          GSF=SILIN(R(I,J-1,K  ),R(I,J,K  ),DYM1,DYJ)*V(I,J  ,K  )
          GSB=SILIN(R(I,J-1,K-1),R(I,J,K-1),DYM1,DYJ)*V(I,J  ,K-1)

          GF =SILIN(R(I,J,K+1),R(I,J,K  ),DZFF,DZF)*W(I,J,K+1)
          GP =SILIN(R(I,J,K-1),R(I,J,K  ),DZB ,DZF)*W(I,J,K  )
          GB =SILIN(R(I,J,K-2),R(I,J,K-1),DZBB,DZB)*W(I,J,K-1)

          GEF=SILIN(R(I+1,J,K  ),R(I,J,K  ),DXP1,DXI)*U(I+1,J,K  )
          GEB=SILIN(R(I+1,J,K-1),R(I,J,K-1),DXP1,DXI)*U(I+1,J,K-1)
          GWF=SILIN(R(I-1,J,K  ),R(I,J,K  ),DXM1,DXI)*U(I  ,J,K  )
          GWB=SILIN(R(I-1,J,K-1),R(I,J,K-1),DXM1,DXI)*U(I  ,J,K-1)

C *** MASS FLOW RATE
          CN=SILIN(GNF,GNB,DZF,DZB)*DZXN
          CS=SILIN(GSF,GSB,DZF,DZB)*DZXS
          CE=SILIN(GEF,GEB,DZF,DZB)*DYZE
          CW=SILIN(GWF,GWB,DZF,DZB)*DYZW
          CF=0.5*(GF+GP)*DXYF
          CB=0.5*(GP+GB)*DXYB

C *** VISCOSITY
          VISF=VIS(I,J,K  )
          VISB=VIS(I,J,K-1)
          VISN=(VIS(I,J+1,K)+VIS(I,J,K)+VIS(I,J+1,K-1)+VIS(I,J,K-1))/4.
          VISS=(VIS(I,J-1,K)+VIS(I,J,K)+VIS(I,J-1,K-1)+VIS(I,J,K-1))/4.
          VISE=(VIS(I+1,J,K)+VIS(I,J,K)+VIS(I+1,J,K-1)+VIS(I,J,K-1))/4.
          VISW=(VIS(I-1,J,K)+VIS(I,J,K)+VIS(I-1,J,K-1)+VIS(I,J,K-1))/4.

          VISN1=ZXOYN*VISN
          VISS1=ZXOYS*VISS
          VISE1=YZOXE*VISE
          VISW1=YZOXW*VISW
          VISF1=XYOZF*VISF
          VISB1=XYOZB*VISB

C *** QUICK SCHEME
          CEP=(ABS(CE)+CE)*DXP1*DXI/(8.*DXE*(DXE+DXW ))
          CEM=(ABS(CE)-CE)*DXP1*DXI/(8.*DXE*(DXE+DXEE))
          CWP=(ABS(CW)+CW)*DXM1*DXI/(8.*DXW*(DXW+DXWW))
          CWM=(ABS(CW)-CW)*DXM1*DXI/(8.*DXW*(DXW+DXE ))

          CNP=(ABS(CN)+CN)*DYP1*DYJ/(8.*DYN*(DYN+DYS ))
          CNM=(ABS(CN)-CN)*DYP1*DYJ/(8.*DYN*(DYN+DYNN))
          CSP=(ABS(CS)+CS)*DYM1*DYJ/(8.*DYS*(DYS+DYSS))
          CSM=(ABS(CS)-CS)*DYM1*DYJ/(8.*DYS*(DYS+DYN ))

          CFP=(ABS(CF)+CF)*DZF/(DZK *16.)
```

```
          CFM=(ABS(CF)-CF)*DZF/(DZP1*16.)
          CBP=(ABS(CB)+CB)*DZB/(DZM1*16.)
          CBM=(ABS(CB)-CB)*DZB/(DZK *16.)

          AE(I,J,K)=(-.5*CE*DXI+CWM*DXW)/DXE+CEP+CEM*(1.+DXE/DXEE)+VISE1
          AW(I,J,K)=( .5*CW*DXI+CEP*DXE)/DXW+CWM+CWP*(1.+DXW/DXWW)+VISW1
          AN(I,J,K)=(-.5*CN*DYJ+CSM*DYS)/DYN+CNP+CNM*(1.+DYN/DYNN)+VISN1
          AS(I,J,K)=( .5*CS*DYJ+CNP*DYN)/DYS+CSM+CSP*(1.+DYS/DYSS)+VISS1
          AF(I,J,K)= -.5*CF    +CBM*DZB /DZF+CFP+CFM*(1.+DZF/DZFF)+VISF1
          AB(I,J,K)=  .5*CB    +CFP*DZF /DZB+CBM+CBP*(1.+DZB/DZBB)+VISB1

C *** BOUNDARY CONSIDERATION
          IF (I.LT.NI+3) THEN
             AEE=-CEM*DXE/DXEE
             AEER=AEE*WPD(I+2,J,K)
          ELSE
             AEE=0.
             AEER=0.
          ENDIF

          IF (I.GT.2) THEN
             AWW=-CWP*DXW/DXWW
             AWWR=AWW*WPD(I-2,J,K)
          ELSE
             AWW=0.
             AWWR=0.
          ENDIF

          IF (J.LT.NJ+3) THEN
             ANN=-CNM*DYN/DYNN
             ANNR=ANN*WPD(I,J+2,K)
          ELSE
             ANN=0.
             ANNR=0.
          ENDIF

          IF (J.GT.2) THEN
             ASS=-CSP*DYS/DYSS
             ASSR=ASS*WPD(I,J-2,K)
          ELSE
             ASS=0.
             ASSR=0.
          ENDIF

          IF (K.LT.NK+3) THEN
             AFF=-CFM*DZF/DZFF
             AFFR=AFF*WPD(I,J,K+2)
          ELSE
             AFF=0.
             AFFR=0.
          ENDIF

          IF (K.GT.3) THEN
             ABB=-CBP*DZB/DZBB
             ABBR=ABB*WPD(I,J,K-2)
          ELSE
```

```
            ABB=0.
            ABBR=0.
         ENDIF

C *** MODIFICATION FOR DECK BOUNDARIES
         IF (NOD(I-1,J,K).NE.0) THEN
            AWW=0.0
            AWWR=0.0
         ENDIF

         IF (NOD(I+1,J,K).NE.0) THEN
            AEE=0.0
            AEER=0.0
         ENDIF

         IF (NOD(I,J-1,K).NE.0) THEN
            ASS=0.0
            ASSR=0.0
         ENDIF

         IF (NOD(I,J+1,K).NE.0) THEN
            ANN=0.0
            ANNR=0.0
         ENDIF

         IF (NOD(I,J,K-2).NE.0) THEN
            ABB=0.0
            ABBR=0.0
         ENDIF

         IF (NOD(I,J,K+1).NE.0) THEN
            AFF=0.0
            AFFR=0.0
         ENDIF

C *** SU FROM NORMAL STRESS
         RF=(SIG33(I,J,K  )-(W(I,J,K+1)-W(I,J,K  ))*VISF/DZF)*DXYF
         RB=(SIG33(I,J,K-1)-(W(I,J,K  )-W(I,J,K-1))*VISB/DZB)*DXYB
         RN=(SIG23(I,J+1,K)-(W(I,J+1,K)-W(I,J  ,K))*VISN/DYN)*DZXN
         RS=(SIG23(I,J  ,K)-(W(I,J  ,K)-W(I,J-1,K))*VISS/DYS)*DZXS
         RE=(SIG13(I+1,J,K)-(W(I+1,J,K)-W(I  ,J,K))*VISE/DXE)*DYZE
         RW=(SIG13(I  ,J,K)-(W(I  ,J,K)-W(I-1,J,K))*VISW/DXW)*DYZW

C *** SU FROM CURVED STRESSES AND ACCELERATIONS
         AVG23= 0.5*(SIG23(I  ,J+1,K)+SIG23(I,J,K))
         AVG13= 0.5*(SIG13(I+1,J  ,K)+SIG13(I,J,K))
         AVG22=SILIN(SIG22(I,J,K),SIG22(I,J,K-1),DZF,DZB)
         AVG11=SILIN(SIG11(I,J,K),SIG11(I,J,K-1),DZF,DZB)

         AU3=W(I,J,K)
         AU2=BILIN(V(I,J+1,K  ),V(I,J,K  ),DYJ,DYJ,
     &             V(I,J+1,K-1),V(I,J,K-1),DYJ,DYJ,DZF,DZB)
         AU1=BILIN(U(I+1,J,K  ),U(I,J,K  ),DXI,DXI,
     &             U(I+1,J,K-1),U(I,J,K-1),DXI,DXI,DZF,DZB)

         AR=SILIN(R(I,J,K),R(I,J,K-1),DZF,DZB)
```

```
          ARU23=AR*AU2*AU3
          ARU13=AR*AU1*AU3
          ARU22=AR*AU2*AU2
          ARU11=AR*AU1*AU1

          RRY=(AVG23-ARU23)*DXI*(DZN-DZS)
          RRX=(AVG13-ARU13)*DYJ*(DZE-DZW)
          RRZ=(AVG22-ARU22)*DXI*(DYF-DYB)+(AVG11-ARU11)*DYJ*(DXF-DXB)

          AP(I,J,K)=AE(I,J,K)+AW(I,J,K)+AN(I,J,K)+AS(I,J,K)
     &              +AF(I,J,K)+AB(I,J,K)+AEE+AWW+ANN+ASS+AFF+ABB
          SP(I,J,K)=-(ROD(I,J,K)*DZB+ROD(I,J,K-1)*DZF)*VOLDT/(DZB+DZF)
          SU(I,J,K)=-SP(I,J,K)*WOD(I,J,K)+DXI*DYJ*(P(I,J,K-1)-P(I,J,K))+
     &              AEER+AWWR+ANNR+ASSR+AFFR+ABBR+RE-RW+RN-RS+RF-RB+RRY+
     &              RRX-RRZ-BUOY*((R(I,J,K)-REQ(I,J,K))*DZB+(R(I,J,K-1)-
     &              REQ(I,J,K-1))*DZF)*VOL/(DZB+DZF)
  100  CONTINUE

C *** TAKE CARE OF B.C. THRU AN,AS,AE,AW,AP AND SU

C *** Y DIRECTION
       DO 500 K=3,NK+3
       DO 500 I=2,NI+3
          SP(I,2   ,K)=SP(I,2   ,K)-AS(I,2   ,K)
          SP(I,NJ+3,K)=SP(I,NJ+3,K)-AN(I,NJ+3,K)
          AS(I,2   ,K)=0.
          AN(I,NJ+3,K)=0.
  500  CONTINUE

C *** X DIRECTION
       DO 502 K=3,NK+3
       DO 502 J=2,NJ+3
          SP(2   ,J,K)=SP(2   ,J,K)-AW(2   ,J,K)
          SP(NI+3,J,K)=SP(NI+3,J,K)-AE(NI+3,J,K)
          AW(2   ,J,K)=0.0
          AE(NI+3,J,K)=0.0
  502  CONTINUE

C *** Z DIRECTION
       DO 600 I=2,NI+3
       DO 600 J=2,NJ+3
          AF(I,J,NK+3)=0.
          AB(I,J,3   )=0.
  600  CONTINUE

C *** MODIFICATION FOR DECK BOUNDARIES
       IF (NCHIP.EQ.0) GOTO 201
       DO 101 N=1,NCHIP
          IB  =ICHPB(N)
          IE  =IB+NCHPI(N)-1
          JB  =JCHPB(N)
          JE  =JB+NCHPJ(N)-1
          KB  =KCHPB(N)
          KE  =KB+NCHPK(N)-1
```

```
          DO 102 J=JB,JE-1
          DO 102 K=KB,KE
             SP(IB-1,J,K)=SP(IB-1,J,K)-AE(IB-1,J,K)
             SP(IE  ,J,K)=SP(IE  ,J,K)-AW(IE  ,J,K)
             SU(IB-1,J,K)=SU(IB-1,J,K)+AE(IB-1,J,K)*WFAN(N)*2.0
             SU(IE  ,J,K)=SU(IE  ,J,K)+AW(IE  ,J,K)*WFAN(N)*2.0
             AE(IB-1,J,K)=0.0
             AW(IE  ,J,K)=0.0
  102     CONTINUE

          DO 103 I=IB,IE-1
          DO 103 K=KB,KE
             SP(I,JB-1,K)=SP(I,JB-1,K)-AN(I,JB-1,K)
             SP(I,JE  ,K)=SP(I,JE  ,K)-AS(I,JE  ,K)
             SU(I,JB-1,K)=SU(I,JB-1,K)+AN(I,JB-1,K)*WFAN(N)*2.0
             SU(I,JE  ,K)=SU(I,JE  ,K)+AS(I,JE  ,K)*WFAN(N)*2.0
             AN(I,JB-1,K)=0.0
             AS(I,JE  ,K)=0.0
  103 CONTINUE

          DO 106 I=IB,IE-1
          DO 106 J=JB,JE-1
             SU(I,J,KB-1)=SU(I,J,KB-1)+AF(I,J,KB-1)*WFAN(N)
             SU(I,J,KE+1)=SU(I,J,KE+1)+AB(I,J,KE+1)*WFAN(N)
             AF(I,J,KB-1)=0.0
             AB(I,J,KE+1)=0.0
  106     CONTINUE

C *** FOR THE CELLS INSIDE OF THE DECKS
          DO 104 I=IB,IE-1
          DO 104 J=JB,JE-1
          DO 104 K=KB,KE
             SP(I,J,K)=-1.0E2
             AW(I,J,K)=0.
             AE(I,J,K)=0.
             AS(I,J,K)=0.
             AN(I,J,K)=0.
             AB(I,J,K)=0.
             AF(I,J,K)=0.
             SU(I,J,K)=1.0E2*WFAN(N)
  104     CONTINUE
  101 CONTINUE

C *** ASSEMBLE COEFFICIENTS AND SOLVE DIFFERENCE EQUATIONS
  201 DO 301 K=3,NK+3
      DO 301 J=2,NJ+3
      DO 301 I=2,NI+3
         DXI=DXXC(I)
         DYJ=DYYC(J)
         DXY=DXI*DYJ
         AP(I,J,K)=AP(I,J,K)-SP(I,J,K)
         DW(I,J,K)=DXY/AP(I,J,K)
  301 CONTINUE

C *** SOLVE FOR W
      CALL TRID  (3,3,4,NI+2,NJ+2,NK+2,W)
```

```
C *** RESET THE VELOCITY INSIDE OF THE DECKS
      IF (NCHIP.EQ.0) GOTO 111
      DO 110 N=1,NCHIP
         IB=ICHPB(N)
         IE=IB+NCHPI(N)-1
         JB=JCHPB(N)
         JE=JB+NCHPJ(N)-1
         KB=KCHPB(N)
         KE=KB+NCHPK(N)-1
         DO 108 I=IB,IE-1
         DO 108 J=JB,JE-1
         DO 108 K=KB,KE
            W(I,J,K)=WFAN(N)
  108    CONTINUE
  110 CONTINUE

  111 RETURN
      END

*******************************************************************************
*******************************************************************************
      SUBROUTINE GLOBE
*******************************************************************************
*THIS SUBROUTINE CALCULATES THE GLOBAL PRESSURE CORRECTION, WHEREBY THE
*PRESSURE MATRIX IS UPDATED.
*
*VARIABLES USED ARE:
* SUMT   = SUM OF TEMPERATURES
* SUMPT  = SUM OF PRESSURE OVER TEMPERATURE
* SUMPET = SUM OF EQUILIBRIUM PRESSURE OVER TEMP
* UGRT   = CONSTANT (FROM SUBROUTINE INIT)
* PCORR  = PRESSURE CORRECTION
*******************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL16/U0,UGRT,BUOY,CP0,PRT,COND0,VIS0,RHO0,
     &            TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
     &            SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
     &            DV(25,25,15),DW(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

      SUMT=0.
      SUMPT=0.
      SUMPET=0.
      DO 370 I=3,NI+2
      DO 370 J=3,NJ+2
      DO 370 K=3,NK+2
         IF (NOD(I,J,K).NE.1) THEN
```

```
            DXI    = DXXC(I)
            DYJ    = DYYC(J)
            DZK    = DZZC(K)
            VOL    = DXI*DYJ*DZK
            SUMT   = SUMT+VOL/T(I,J,K)
            SUMPT  = SUMPT+P(I,J,K)*VOL/T(I,J,K)
            SUMPET = SUMPET+REQ(I,J,K)*VOL*(1.-1./T(I,J,K))
         ENDIF
  370 CONTINUE
      SUMPET = SUMPET/UGRT
      PCORR  = (SUMPET-SUMPT)/SUMT

      DO 371 I=1,NI+4
      DO 371 J=1,NJ+4
      DO 371 K=1,NK+4
         P(I,J,K) = P(I,J,K)+PCORR
  371 CONTINUE

      RETURN
      END

*******************************************************************************
*******************************************************************************
      SUBROUTINE GRID
*******************************************************************************
*NONDIMENSIONAL VARIABLES:
*  GRID SIZES:
*    DX = X DIRECTION
*    DY = Y DIRECTION
*    DZ = Z DIRECTION
*
*  CENTRAL CELLS:
*    XC()   = X COORDINATE
*    YC()   = Y COORDINATE
*    ZC()   = Z COORDINATE
*    DXXC() = X LENGTH
*    DYYC() = Y LENGTH
*    DZZC() = Z LENGTH
*
*  STAGGERED CELLS:
*    XS()   = X COORDINATE
*    YS()   = Y COORDINATE
*    ZS()   = Z COORDINATE
*    DXXS() = X LENGTH
*    DYYS() = Y LENGTH
*    DZZS() = Z LENGTH
*
*******************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL2/X,Y,H,TFLR,TWAL
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
```

```
C *** GENERATION OF THE GRIDS
      DX=X/(DFLOAT(NI)*H)
      DY=Y/(DFLOAT(NJ)*H)
      DZ=H/(DFLOAT(NK)*H)

C *** CALCULATE XS,YS,ZS (COORDINATES OF STAGGERED CV'S)
      DO 10 I=3,NI+3
         XS(I)=(I-3)*DX
   10 CONTINUE
      XS(2)=XS(3)-TWAL/(H*12.)
      XS(1)=XS(2)-TWAL/(H*12.)
      XS(NI+4)=XS(NI+3)+TWAL/(H*12.)
      XS(NI+5)=XS(NI+4)+TWAL/(H*12.)

      DO 12 J=3,NJ+3
         YS(J)=(J-3)*DY
   12 CONTINUE
      YS(2)=YS(3)-TWAL/(H*12.)
      YS(1)=YS(2)-TWAL/(H*12.)
      YS(NJ+4)=YS(NJ+3)+TWAL/(H*12.)
      YS(NJ+5)=YS(NJ+4)+TWAL/(H*12.)

      DO 14 K=3,NK+3
         ZS(K)=(K-3)*DZ
   14 CONTINUE
      ZS(2)=ZS(3)-TFLR/(H*12.)
      ZS(1)=ZS(2)-TFLR/(H*12.)
      ZS(NK+4)=ZS(NK+3)+TFLR/(H*12.)
      ZS(NK+5)=ZS(NK+4)+TFLR/(H*12.)

C *** CALCULATE DXXC,DYYC AND DZZC (DIMENSIONS OF CENTERED CV'S)
      DO 20 I=1,NI+4
         DXXC(I)=XS(I+1)-XS(I)
   20 CONTINUE
      DXXC(NI+5)=DXXC(NI+4)

      DO 22 J=1,NJ+4
         DYYC(J)=YS(J+1)-YS(J)
   22 CONTINUE
      DYYC(NJ+5)=DYYC(NJ+4)

      DO 24 K=1,NK+4
         DZZC(K)=ZS(K+1)-ZS(K)
   24 CONTINUE
      DZZC(NK+5)=DZZC(NK+4)

C *** CALCULATE DXXS,DYYS,DZZS (DIMENSIONS OF STAGGERED CV'S)
      DO 30 I=2,NI+5
         DXXS(I)=(DXXC(I)+DXXC(I-1))/2.0
   30 CONTINUE
      DXXS(1)=DXXS(2)

      DO 32 J=2,NJ+5
         DYYS(J)=(DYYC(J)+DYYC(J-1))/2.0
   32 CONTINUE
      DYYS(1)=DYYS(2)
```

```
      DO 34 K=2,NK+5
         DZZS(K)=(DZZC(K)+DZZC(K-1))/2.0
   34 CONTINUE
      DZZS(1)=DZZS(2)

C *** CALCULATE XC,YC,ZC (LOCATION OF CENTER CELLS)
      DO 40 I=1,NI+5
         XC(I)=XS(I)+DXXC(I)/2.0
   40 CONTINUE

      DO 42 J=1,NJ+5
         YC(J)=YS(J)+DYYC(J)/2.0
   42 CONTINUE

      DO 44 K=1,NK+5
         ZC(K)=ZS(K)+DZZC(K)/2.0
   44 CONTINUE

      RETURN
      END

*******************************************************************************
*******************************************************************************
      SUBROUTINE INIT
*******************************************************************************
*THIS SUBROUTINE INITIALIZES THE FIELD AND CONSTANTS WITH RESPECT
*TO INITIAL START OR RESTARTING CAPABILITY.
*
*VARIABLES ARE :
* ALEW         = LEWIS NUMBER (USED IN SMOKE CONCENTRATION CALCULATIONS)
* BUOY         = BUOYANCY FORCE CONSTANT
* C0           = INITIAL SMOKE CONCENTRATION
* CONDO        = REFERENCE CONDUCTIVITY
* CONSRA       = NONDIMENSIONAL RADIATION CONSTANT
* CPO          = REFERENCE SPECIFIC HEAT
* F            = INITIAL MASS OF FUEL (LBM)
* FR           = MASS OF FUEL REMAINING (LBM)
* GC           = GRAVITY CONSTANT
* H            = CHARACTERISTIC LENGTH;HEIGHT OF CHAMBER=10.FT
* HCOEF        = DIMENSIONLESS HEAT TRANSFER COEF
* HCONV        = HEAT TRANSFER COEFFICIENT IN BTU/(HR*FT**2*DEGREES)
* HR           = HEIGHT IN CM
* NTAPE        = NONDIMENSIONAL FORMS OF TTAPE
* NWRITE       = NONDIMENSIONAL FORMS OF TWRITE
* RHO0         = REFERENCE DENSITY
* TA           = TEMP IN DEGREES RANKINE
* TIME         = DIMENSIONLESS TIME
* TR           = TEMP IN DEGREES KELVIN
* U0           = CHARACTERISTIC VELOCITY (1 FT/SEC)
* UGRT         = PERFECT GAS LAW NONDIMENSIONAL CONSTANT
* VISO         = REFERENCE VISCOSITY (NONDIM)
* VISL         = MINIMUM VISCOSITY (NONDIM)
* VISMAX       = MAXIMUM VISCOSITY (NONDIM)
*
*MATRICES OF THE FORM
```

```
* _OD          = DIMENSIONLESS PARAMETER AT PREVIOUS TIME STEP
* _            = DIMENSIONLESS PARAMETER AT CURRENT TIME STEP
* _PD          = DIMENSIONLESS PARAMETER AT NEXT TIME STEP
*
*WHERE THE PARAMETERS ARE
* AP           = COEFICIENT AT NODE P
* AE,AW,AN     = COEFICIENTS AT PTS EAST,WEST,NORTH,
* AS,AF,AB             SOUTH, FRONT, AND BACK
* CPM          = MEAN SPECIFIC HEAT
* COND( )      = CONDUCTIVITY MATRIX
* CX,CY,CZ     = LOCATION OF THERMOCOUPLE IN X,Y,Z
* DU,DV,DW     = USED IN PRESSURE CORRECTION SUBROUTINE
* DXXC,DYYC    = LENGTH AROUND THE CENTER CELL
* DZZC
* DXXS,DYYS    = LENGTH AROUND THE STAGGERED CELL
* DZZS
* NOD          = IF EQUAL TO ZERO, LIQUID; IF EQUAL TO ONE, SOLID
* PP           = CORRECTED PRESSURE (P')
* REQ          = DENSITY AT EQUILIBRIUM
* SMP          = RESIDUAL MASS SUMMATION OF NODAL POINT
* SMPP         = LENGTH SCALE FOR TURBULENCE
* SP           = BOUNDARY CONDITION TERM AT NODE P
* SU           = SOURCE TERM
* T,P,C        = TEMP, PRESSURE, AND SMOKE CONCENTRATION
* U,V,W        = VELOCITY COMPONENTS IN X,Y,X DIRECTIONS
* VIS          = VISCOSITY
* _B,_E        = BEGINNING AND ENDING NODAL POINT FOR
*                            THE SOLID IN I,J,K
* XC,YC,ZC     = X,Y,Z LOCATION OF CENTER CELL NODAL POINT
* XS,YS,ZS     = X,Y,Z LOCATION OF STAGGERED CELL NODAL POINT
**************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL2/X,Y,H,TFLR,TWAL
      COMMON/BL3/F,FR,HSTART
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL12/NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL14/HCOEF,CNT,ABTURB,BTURB,VISL,VISMAX
      COMMON/BL16/U0,UGRT,BUOY,CP0,PRT,COND0,VIS0,RHO0,
     &            TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
      COMMON/BL20/SIG11(25,25,15),SIG12(25,25,15),SIG22(25,25,15),
     &            SIG13(25,25,15),SIG23(25,25,15),SIG33(25,25,15)
      COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
     &            JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
      COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
     &            COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
     &            WOD(25,25,15)
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL33/TPD(25,25,15),RPD(25,25,15),PPD(25,25,15),
     &            CPD(25,25,15),UPD(25,25,15),VPD(25,25,15),
     &            WPD(25,25,15)
      COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
```

```
     &            SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
     &            DV(25,25,15),DW(25,25,15)
      COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
     &            AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
     &            SU(25,25,15),RI(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
      COMMON/BL38/TCOUP(30),CX(30),CY(30),CZ(30),NTH(30,3),NTHCO
      COMMON/BL39/ALEW,CONSRA,QSIN,QSWER,QSWAL,QSAIR,QSFAN
      COMMON/BL43/QSCONF,QSCONB,QSCONE,QSCONW,QSCONN,QSCONS,
     &            QSRADF,QSRADB,QSRADE,QSRADW,QSRADN,QSRADS,
     &            WAIR,WWAL,WINS,WERR,WWFAN

C *** INITIALIZE GIVEN PARAMETERS
      C0=0.0
      F=200.0
      HCONV=15.0
      NBLOR=13
      PI=4.*ATAN(1.)
      TCOOL=1.0

C *** NONDIMENSIONALIZE THE REFERENCE VISCOSITY
      VIS0=VIS0/(U0*H)
      VISL=VIS0

C *** SET MAXIMUM VISCOSITY
      VISMAX=400.*VISL

C *** NONDIMENSIONALIZE THE HEAT TRANSFER COEFFICIENT
      HCOEF=HCONV/(3600.*CP0*RHO0*U0)

      COND0=VIS0/PRT
      FR=F
      BUOY=GC*H/(U0**2)
      UGRT=U0**2/(GC*RAIR*TA)
      CONSRA=1.714E-9*TA**3/(RHO0*CP0*U0*3600.)

C *** FOR ENERGY DISTRIBUTION
      QSIN=0.
      QSWER=0.
      QSWAL=0.
      QSAIR=0.
      QSFAN=0.

C *** INITIALIZE CONDUCTION HEAT FLUX TO EACH WALL
      QSCONF=0.
      QSCONB=0.
      QSCONE=0.
      QSCONW=0.
      QSCONN=0.
      QSCONS=0.

C *** INITIALIZE RADIATION HEAT FLUX TO EACH WALL
      QSRADF=0.
      QSRADB=0.
      QSRADE=0.
```

```
      QSRADW=0.
      QSRADN=0.
      QSRADS=0.

      NWRITE=TWRITE*U0/(DTIME*H)
      NTAPE=TTAPE*U0/(DTIME*H)

C *** INITIALIZE VARIABLE FIELDS
      DO 220 J=1,NJ+4
      DO 220 I=1,NI+4
      DO 220 K=1,NK+4

         IF(KRUN.LE.0) THEN
            UOD(I,J,K)  =0.
            VOD(I,J,K)  =0.
            WOD(I,J,K)  =0.
            POD(I,J,K)  =0.
            TOD(I,J,K)  =TA/TA
            COD(I,J,K)  =C0
         ENDIF

         U(I,J,K)    =UOD(I,J,K)
         UPD(I,J,K)  =UOD(I,J,K)

         V(I,J,K)    =VOD(I,J,K)
         VPD(I,J,K)  =VOD(I,J,K)

         W(I,J,K)    =WOD(I,J,K)
         WPD(I,J,K)  =WOD(I,J,K)

         P(I,J,K)    =POD(I,J,K)
         PPD(I,J,K)  =POD(I,J,K)

         T(I,J,K)    =TOD(I,J,K)
         TPD(I,J,K)  =TOD(I,J,K)

         C(I,J,K)    =COD(I,J,K)
         CPD(I,J,K)  =COD(I,J,K)

         DU(I,J,K)   =0.
         DV(I,J,K)   =0.
         DW(I,J,K)   =0.

         SU(I,J,K)   =0.
         SP(I,J,K)   =0.

         PP(I,J,K)   =0.

         AP(I,J,K)   =0.
         AW(I,J,K)   =0.
         AE(I,J,K)   =0.
         AN(I,J,K)   =0.
         AS(I,J,K)   =0.
         AF(I,J,K)   =0.
         AB(I,J,K)   =0.
```

```
         SMP(I,J,K)  =0.
         SMPP(I,J,K) =0.

         SIG11(I,J,K)=0.
         SIG12(I,J,K)=0.
         SIG13(I,J,K)=0.
         SIG22(I,J,K)=0.
         SIG23(I,J,K)=0.
         SIG33(I,J,K)=0.

         VIS(I,J,K)  =VISL
         COND(I,J,K) =COND0
         CPM(I,J,K)  =1.0E0
         NOD(I,J,K)  =0
  220 CONTINUE

C *** DEFINE THERMAL PROPERTIES OF DECK AND SOLID
      IF (NCHIP.NE.0) CALL SOLCON

C *** DEFINE HEIGHT OF NODE POINTS AND COMPUTE HYDROSTATIC
C     EQUILIBRIUM DENSITY REQ(I,J,K)
   15 DO 229 J=1,NJ+4
      DO 229 I=1,NI+4
      DO 229 K=1,NK+4
         HEIGHT(I,J,K)=ZC(K)
         REQ(I,J,K)   =EXP(-BUOY*UGRT*HEIGHT(I,J,K))
         IF(KRUN.LE.0) THEN
            ROD(I,J,K)=REQ(I,J,K)/TPD(I,J,K)
         ENDIF
         R(I,J,K)  =ROD(I,J,K)
         RPD(I,J,K)=ROD(I,J,K)
  229 CONTINUE

C *** FOLLOWING IS FOR DETERMINING THE THERMOCOUPLE POSITIONS
      DO 5000 N=1,NTHCO
         DO 5001 I=1,NI+4
            IF (XC(I).LT.CX(N).AND.XC(I+1).GE.CX(N)) GOTO 5002
 5001    CONTINUE

 5002    II=I
         DO 5003 J=1,NJ+4
            IF (YC(J).LT.CY(N).AND.YC(J+1).GE.CY(N)) GOTO 5004
 5003    CONTINUE

 5004    JJ=J
         DO 5005 K=1,NK+4
            IF (ZC(K).LT.CZ(N).AND.ZC(K+1).GE.CZ(N)) GOTO 5006
 5005    CONTINUE

 5006    KK=K
         NTH(N,1)=II
         NTH(N,2)=JJ
         NTH(N,3)=KK
 5000 CONTINUE

      RETURN
```

```
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE INPUT(NSTOP)
C*********************************************************************
*THIS SUBROUTINE SETS UP REQUIRED VALUES TO BEGIN THE PROGRAM.
*
*VARIABLES ARE:
*  KRUN       = RESTART INDICATOR
*  NCHIP      = NUMBER OF INTERNAL SOLID PIECES
*  NMS        = NUMBER OF MASS SOURCES
*  NWRP       = NUMBER OF TIME STEPS BETWEEN WRITES TO OUTPUT FILE
*  NTHCO      = NUMBER OF THERMOCOUPLES TO PRINT OUT
*  TMAX       = NONDIMENSIONAL MAXIMUM TIME ALLOWED
*  XTMAX      = MAXIMUM TIME ALLOWED (SECONDS)
*  TWRITE     = TIME BETWEEN FIELD VARIABLE OUTPUT (SECONDS)
*  TTAPE      = TIME INTERVAL BETWEEN PLOTS (SECONDS)
*  DTIME      = NONDIMENSIONAL TIME STEP
*  XDTIME     = TIME STEP (SECONDS)
*  HSTART     = FIRE START TIME (SECONDS)
*  NHSZ(1,1) = STARTING NODE OF HEAT SOURCE, X-DIR
*  NHSZ(2,1) =                               Y-DIR
*  NHSZ(3,1) =                               Z-DIR
*  NHSZ(1,2) = ENDING NODE OF HEAT SOURCE, X-DIR
*  NHSZ(2,2) =                             Y-DIR
*  NHSZ(3,2) =                             Z-DIR
*  ICHPB      = FIRST NODE OF INTERNAL SOLID IN X DIR
*  JCHPB      =                                 Y DIR
*  KCHPB      =                                 Z DIR
*  NCHPI      = NUMBER OF INTERNAL SOLID NODES IN X DIR
*  NCHPJ      =                                   Y DIR
*  NCHPK      =                                   Z DIR
*  IMSB       = FIRST MASS SOURCE NODE IN X DIR
*  JMSB       =                           Y DIR
*  KMSB       =                           Z DIR
*  NMSI       = NUMBER OF MASS SOURCE NODES IN X DIR
*  NMSJ       =                               Y DIR
*  NMSK       =                               Z DIR
*  RMS        = DIMENSIONLESS MASS SOURCE
*               (= CFM/(60.*H**2*U0*NMSI*NMSJ*NMSK)
*  CX,CY,CZ   = THERMOCOUPLE POSITIONS IN X,Y,Z
C*********************************************************************
*DATA FILES USED IN THIS PROGRAM:
*
* FILE # 10 = FIRE.DAT  : INITIAL SET-UP DATA
*        11 = FIRE1.CONT : RESTART/CONTINUATION DATA
*
C*********************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL2/X,Y,H,TFLR,TWAL
      COMMON/BL3/F,FR,HSTART
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL12/NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
```

```
      COMMON/BL16/U0,UGRT,BUOY,CP0,PRT,COND0,VIS0,RHO0,
     &            TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
      COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
     &            JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
      COMMON/BL23/RMS(20),NMS,IMSB(20),NMSI(20),JMSB(20),NMSJ(20),
     &            KMSB(20),NMSK(20)
      COMMON/BL31/TOD(25,25,15),ROD(25,25,15),POD(25,25,15),
     &            COD(25,25,15),UOD(25,25,15),VOD(25,25,15),
     &            WOD(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
      COMMON/BL38/TCOUP(30),CX(30),CY(30),CZ(30),NTH(30,3),NTHCO

      CHARACTER ANS*1
      LOGICAL L1,L2
      NSTOP=0
      KRUN=0

***** CHECK FOR INPUT DATA FILE
      INQUIRE (FILE='/FIRE DATA B1',EXIST=L1)
      IF (L1) THEN

C *** READ IN DATA FROM EXISTING DATA FILE
         OPEN(10,FILE='/FIRE DATA B1',STATUS='OLD')
         REWIND(10)
         READ(10,*) X,Y,H,TFLR,TWAL,TA
         READ(10,*) NI,NJ,NK
         READ(10,*) NCHIP,NMS,NWRP,NTHCO
         READ(10,*) TMAX,DTIME,TTAPE,TWRITE,HSTART
         READ(10,*) NHSZ(1,1),NHSZ(1,2),NHSZ(2,1),NHSZ(2,2),NHSZ(3,1),
     &              NHSZ(3,2)
         IF (NCHIP.LE.0) GOTO 33
         DO 32 N=1,NCHIP
            READ(10,*) ICHPB(N),NCHPI(N),JCHPB(N),NCHPJ(N),KCHPB(N),
     &                 NCHPK(N),CPS(N),CONS(N),WFAN(N)
   32    CONTINUE
   33    IF (NMS.LE.0) GOTO 37
         DO 36 N=1,NMS
            READ(10,*) IMSB(N),NMSI(N),JMSB(N),NMSJ(N),KMSB(N),
     &                 NMSK(N),RMS(N)
   36    CONTINUE
   37    DO 38 I=1,NTHCO
            READ (10,*) CX(I),CY(I),CZ(I)
   38    CONTINUE
         REWIND(10)
         CLOSE(10)
      ELSE

C *** STOP PROGRAM IF INPUT DATA NOT AVAILABLE
         NSTOP=9999
         GOTO 999
      ENDIF

***** CHECK FOR CONTINUATION FILE
      INQUIRE (FILE='/CONTINUE DATA B4',EXIST=L2)
      IF (L2) THEN
```

```
C *** READ IN DATA FROM OLD CONTINUATION FILE
         OPEN(11,FILE='/CONTINUE DATA B4',STATUS='OLD',
     &           FORM='UNFORMATTED')
         KRUN=1
         REWIND(11)
         READ(11) TIME,NTMAX0,FR,TOD,ROD,UOD,VOD,WOD,POD,COD
         REWIND(11)
         IF(TIME.GE.TMAX) TMAX=TIME+TMAX
       ELSE

C *** CREATE NEW CONTINUATION FILE
         OPEN(11,FILE='/CONTINUE DATA B4',STATUS='NEW',
     &           FORM='UNFORMATTED')
         KRUN=0
       ENDIF

  999 RETURN
      END

*************************************************************************
*************************************************************************
      SUBROUTINE OUT(NN)
*************************************************************************
*THIS SUBROUTINE GENERATES OUTPUT.
*
* NN = 1     SELECTED VALUES ARE PRINTED. INCLUDING TIME, ERROR,
*            PRESSURE, HEAT GENERATION
* NN = 2     TEMPERATURE AT THE THERMOCOUPLES
* NN = 3     FILED VALUES ARE PRINTED
* NN = 4     ENERGY DISTRIBUTION
*************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL2/X,Y,H,TFLR,TWAL
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL12/NWRITE,NTAPE,NTMAX0,NTREAL,TIME,SORSUM,ITER
      COMMON/BL16/U0,UGRT,BUOY,CP0,PRT,COND0,VIS0,RHO0,
     &            TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL34/HEIGHT(25,25,15),REQ(25,25,15),SMP(25,25,15),
     &            SMPP(25,25,15),PP(25,25,15),DU(25,25,15),
     &            DV(25,25,15),DW(25,25,15)
      COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
     &            AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
     &            SU(25,25,15),RI(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
      COMMON/BL38/TCOUP(30),CX(30),CY(30),CZ(30),NTH(30,3),NTHCO
      COMMON/BL39/ALEW,CONSRA,QSIN,QSWER,QSWAL,QSAIR,QSFAN
      COMMON/BL43/QSCONF,QSCONB,QSCONE,QSCONW,QSCONN,QSCONS,QSRADF,
     &            QSRADB,QSRADE,QSRADW,QSRADN,QSRADS,WAIR,WWAL,WINS,
     &            WERR,WWFAN
```

```
*************************************************************************

  500 FORMAT(1X,'TIME=',F7.3,' SECONDS',2X,'NTREAL=',I5,2X,'ITER=',I5,
     &       2X,'SOURCE=',F9.6,2X,'SORSUM=',F9.6,2X,' Q(KW) = ',F10.4)
  504 FORMAT(/,1X,'J=',I2,5X,'K=',I2,/,10X,'T (C)',
     &         4X,'U(CM/SEC)',2X,'V(CM/SEC)',2X,'W(CM/SEC)',/)
  511 FORMAT(1X,'I=',I3,2X,F9.4,2X,F9.4,2X,F9.4,2X,F9.4)
 1084 FORMAT(6X,'*** AT TIME = ',F9.4,' ***',/,
     &       9X,'THE WATTAGE INPUT: WINS = ',E10.4,/,
     &       9X,'WATTAGE INTO AIR:  WAIR = ',E10.4,/,
     &       9X,'WATTAGE DUE TO ERRO     = ',E10.4,/,
     &       9X,'WATTAGE THRU DUCT: WAIR = ',E10.4,/,
     &       9X,'INTO THE WALL:     WWAL = ',E10.4,
     &       2X,'BY CONDUCTION: QTCON = ',E10.4,
     &       2X,'BY RADIATION:  QTRAD = ',E10.4,//)
 2728 FORMAT(9X,'QSRADW: INTO WEST  WALL BY RADIATION= ',E10.4,
     &       9X,'QSCONW: BY CONDUCTION = ',E10.4,/,
     &       9X,'QSRADE: INTO EAST  WALL BY RADIATION= ',E10.4,
     &       9X,'QSCONE: BY CONDUCTION = ',E10.4,/,
     &       9X,'QSRADS: INTO SOUTH WALL BY RADIATION= ',E10.4,
     &       9X,'QSCONS: BY CONDUCTION = ',E10.4,/,
     &       9X,'QSRADN: INTO NORTH WALL BY RADIATION= ',E10.4,
     &       9X,'QSCONN: BY CONDUCTION = ',E10.4,/,
     &       9X,'QSRADB: INTO BACK  WALL BY RADIATION= ',E10.4,
     &       9X,'QSCONB: BY CONDUCTION = ',E10.4,/,
     &       9X,'QSRADF: INTO FRONT WALL BY RADIATION= ',E10.4,
     &       9X,'QSCONF: BY CONDUCTION = ',E10.4,/)
 1088 FORMAT(9X,'PAIR : LOSS INTO CAVITY AIR  = ',F8.3,' %',/,
     &       9X,'PWER : LOSS DUE TO THE ERR   = ',F8.3,' %',/,
     &       9X,'PWALL: LOSS INTO THE WALLS   = ',F8.3,' %',/,
     &       9X,'PFAN : LOSS THROUGH DUCT     = ',F8.3,' %',/,
     &       9X,'PSAIR: TOTAL INTO CAVITY AIR = ',F8.3,' %',/,
     &       9X,'PSWER: TOTAL DUE TO THE ERRO = ',F8.3,' %',/,
     &       9X,'PSFAN: TOTAL THROUGH DUCT    = ',F8.3,' %',/,
     &       9X,'PSWAL: TOTAL INTO THE WALLS  = ',F8.3,' %',//)
 1091 FORMAT(9X,'QSIN  : TOTAL ENERGY INPUT           = ',E10.4,/,
     &       9X,'QSWER : TOTAL ENERGY DUE TO THE ERRO= ',E10.4,/,
     &       9X,'QSAIR : TOTAL ENERGY INTO CAVITY AIR= ',E10.4,/,
     &       9X,'QSFAN : TOTAL ENERGY THROUGH DUCT   = ',E10.4,/,
     &       9X,'QSWAL : TOTAL ENERGY INTO WALLS     = ',E10.4,2X,//)
*************************************************************************

C *** REFERENCE TEMPERATURE IN DEGREES K
      TR=TA/1.8

C *** REFERENCE VELOCITY IN CM/SEC
      UR=U0*30.48

C *** REFERENCE LENGTH IN CM
      HR=H*30.48

      XTIME=TIME*H/U0
      IF (NN.EQ.1) THEN
         QRR=60.**2*QR/3412.
         QKW=60.**2*Q /3412.
         WRITE(12,500) XTIME,NTREAL,ITER,RESORM(ITER),SORSUM,QKW
```

```
      ELSE IF (NN.EQ.2) THEN
         WRITE (12,*)
         WRITE (12,*)' TEMPERATURES  AT THERMOCOUPLE POSITION IN (C):',
     &                (TCOUP(N)*TR-273.16,N=1,NTHCO)
         WRITE (12,*)
         WRITE (12,*)
      ELSE IF (NN.EQ.3) THEN
         WRITE(12,'(1X,A,F10.6)') 'TIME =',XTIME
         DO 501 J=3,NJ+3,NJ
            DO 502 K=2,NK+4
               WRITE(12,504) J,K
  513          DO 503 I=1,NI+4
                  IF (T(I,J,K).LT.TCOOL) T(I,J,K)=TCOOL
                  XTEMP=T(I,J,K)*TR-273.16
                  XR   =1000.*(0.0328)**3*R(I,J,K)*RHO0/2.2048
                  XU   =U(I,J,K)*UR
                  XV   =V(I,J,K)*UR
                  XW   =W(I,J,K)*UR
                  XP   =P(I,J,K)*RHO0*U0**2/(GC*14.696*144.)+REQ(I,J,K)
                  XVIS =VIS(I,J,K)*HR*UR
                  XCOND=COND(I,J,K)*HR*UR
                  WRITE(12,511)I,XTEMP,XU,XV,XW
  503          CONTINUE
  502       CONTINUE
  501    CONTINUE
      ELSE

C *** CALCULATE THE PERCENTAGE AND PRINT OUT THE RESULTS
         QTCON=QSCONF+QSCONB+QSCONS+QSCONN+QSCONW+QSCONE
         QTRAD=QSRADF+QSRADB+QSRADS+QSRADN+QSRADW+QSRADE

C *** WATT PERCENTAGE
         IF (WINS.EQ.0.) WINS=1.0E-5
         PAIR=100.*WAIR /WINS
         PWAL=100.*WWAL /WINS
         PFAN=100.*WWFAN/WINS
         PWER=100.*WERR /WINS

C *** ENERGY PERCENTAGE
         IF (QSIN.EQ.0.0) QSIN=1.0E-3
         PSAIR=100.*QSAIR/QSIN
         PSWAL=100.*QSWAL/QSIN
         PSFAN=100.*QSFAN/QSIN
         PSWER=100.*QSWER/QSIN
         WRITE(12,1084)XTIME,WINS,WAIR,WERR,WWFAN,WWAL,QTCON,QTRAD
         WRITE(12,2728)QSRADW,QSCONW,QSRADE,QSCONE,QSRADS,QSCONS,
     &                 QSRADN,QSCONN,QSRADB,QSCONB,QSRADF,QSCONF
         WRITE(12,1088)PAIR,PWER,PWAL,PFAN,PSAIR,PSWER,PSFAN,PSWAL
         WRITE(12,1091)QSIN,QSWER,QSAIR,QSFAN,QSWAL
      ENDIF

      RETURN
      END
*******************************************************************************
*******************************************************************************
```

```
      SUBROUTINE RADHT (NN)
*************************************************************************
*  NN = CONTROL PARAMETER FOR HEAT FLUX CALCULATIONS
*       WHERE NN=1 :CALCULATE HEAT FLUX FROM FIRE TO WALLS
*             NN=2 :CALCULATE HEAT FLUX FROM FIRE TO BLOCKS
*
*  NTHS         = TOTAL NUMBER OF CV'S CONTAINING HEAT SOURCES
*  NFX,NFY,NFZ = CV NUMBER OF HEAT SOURCE
*  FX,FY,FZ     = COORDINATES OF HEAT SOURCE
*************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &     DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
     &     JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &     U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
     &     AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
     &     SU(25,25,15),RI(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &     CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
      COMMON/BL39/ALEW,CONSRA,QSIN,QSWER,QSWAL,QSAIR,QSFAN
      COMMON/BL40/VFHSW(5,25,15),VFHSE(5,25,15),VFHSS(5,15,25),
     &     VFHSN(5,15,25),VFHSB(5,25,25),VFHSF(5,25,25)
      COMMON/BL41/VFHSBW(5,8,34,34),VFHSBE(5,8,34,34),VFHSBS(5,8,34,34),
     &     VFHSBN(5,8,34,34),VFHSBB(5,8,34,34),VFHSBF(5,8,34,34)

      NTHS=NHSZ(3,2)-NHSZ(3,1)+1
      DO 500 N=1,NTHS
         NFX=NHSZ(1,1)
         NFY=NHSZ(2,1)
         NFZ=NHSZ(3,1)-1+N

C *** AREA OF THE FIRE ELEMENTS
         DYHS=DYYC(NFY)
         DZHS=DZZC(NFZ)
         DAHS=PI*DYHS*DZHS
         EMIS=0.6

C *** NN=1:  CALCULATE RADIATION HEAT FLUX FROM FIRE TO WEST AND
C ***         EAST SURFACES OF THE ENCLOSURE
         IF (NN.EQ.1) THEN
            DO 100 J=3,NJ+2
            DO 100 K=3,NK+2
               SU(2,J,K)   =SU(2,J,K)+CONSRA*EMIS*DAHS*VFHSW(N,J,K)*
     &                      (T(NFX,NFY,NFZ)**4-T(2,J,K)**4)
               SU(NI+3,J,K)=SU(NI+3,J,K)+CONSRA*EMIS*DAHS*VFHSE(N,J,K)*
     &                      (T(NFX,NFY,NFZ)**4-T(NI+3,J,K)**4)
  100       CONTINUE

C *** CALCULATE RADIATION HEAT FLUX FROM FIRE TO NORTH AND
C *** SOUTH SURFACES OF THE ENCLOSURE
```

```
            DO 200 I=3,NI+2
            DO 200 K=3,NK+2
               SU(I,2,K)   =SU(I,2,K)+CONSRA*EMIS*DAHS*VFHSS(N,K,I)*
     &                      (T(NFX,NFY,NFZ)**4-T(I,2,K)**4)
               SU(I,NJ+3,K)=SU(I,NJ+3,K)+CONSRA*EMIS*DAHS*VFHSN(N,K,I)*
     &                      (T(NFX,NFY,NFZ)**4-T(I,NJ+3,K)**4)
  200       CONTINUE

C *** CALCULATE RADIATION HEAT FLUX FROM FIRE TO BACK AND
C *** FRONT SURFACES OF THE ENCLOSURE
            DO 300 I=3,NI+2
            DO 300 J=3,NJ+2
               SU(I,J,2)   =SU(I,J,2)+CONSRA*EMIS*DAHS*VFHSB(N,I,J)*
     &                      (T(NFX,NFY,NFZ)**4-T(I,J,2)**4)
               SU(I,J,NK+3)=SU(I,J,NK+3)+CONSRA*EMIS*DAHS*VFHSF(N,I,J)*
     &                      (T(NFX,NFY,NFZ)**4-T(I,J,NK+3)**4)
  300       CONTINUE
         ENDIF

C *** NN=2:  CALCULATE RADIATION HEAT FLUX FROM FIRE TO WEST AND
C ***         EAST SURFACES OF BLOCK M
         IF (NN.EQ.2) THEN
            IF (NCHIP.LT.NBLOR) THEN
               DO 900 M=1,NCHIP-NBLOR+1
                  IB=ICHPB(M+NBLOR-1)
                  IE=IB+NCHPI(M+NBLOR-1)-1
                  JB=JCHPB(M+NBLOR-1)
                  JE=JB+NCHPJ(M+NBLOR-1)-1
                  KB=KCHPB(M+NBLOR-1)
                  KE=KB+NCHPK(M+NBLOR-1)-1

C *** CALCULATE RADIATION HEAT FLUX FROM FIRE TO WEST AND
C *** EAST SURFACES OF THE BLOCK
                  DO 400 J=JB,JE-1
                  DO 400 K=KB,KE-1
                     SU(IB,J,K)  =SU(IB,J,K)+CONSRA*EMIS*DAHS*
     &                            VFHSBW(N,M,J,K)*(T(NFX,NFY,NFZ)**4-
     &                            T(IB,J,K)**4)
                     SU(IE-1,J,K)=SU(IE-1,J,K)+CONSRA*EMIS*DAHS*
     &                            VFHSBE(N,M,J,K)*(T(NFX,NFY,NFZ)**4-
     &                            T(IE-1,J,K)**4)
  400             CONTINUE

C *** CALCULATE RADIATION HEAT FLUX FROM FIRE TO NORTH AND
C *** SOUTH SURFACES OF THE BLOCK
                  DO 600 I=IB,IE-1
                  DO 600 K=KB,KE-1
                     SU(I,JB,K)  =SU(I,JB,K)+CONSRA*EMIS*DAHS*
     &                            VFHSBS(N,M,K,I)*(T(NFX,NFY,NFZ)**4-
     &                            T(I,JB,K)**4)
                     SU(I,JE-1,K)=SU(I,JE-1,K)+CONSRA*EMIS*DAHS*
     &                            VFHSBN(N,M,K,I)*(T(NFX,NFY,NFZ)**4-
     &                            T(I,JE-1,K)**4)
  600             CONTINUE
```

```
C *** CALCULATE RADIATION HEAT FLUX FROM FIRE TO BACK AND
C *** FRONT SURFACES OF BLOCK
                  DO 700 I=IB,IE-1
                  DO 700 J=JB,JE-1
                     SU(I,J,KB)  =SU(I,J,KB)+CONSRA*EMIS*DAHS*
     &                              VFHSBB(N,M,I,J)*(T(NFX,NFY,NFZ)**4-
     &                              T(I,J,KB)**4)
                     SU(I,J,KE-1)=SU(I,J,KE-1)+CONSRA*EMIS*DAHS*
     &                              VFHSBF(N,M,I,J)*(T(NFX,NFY,NFZ)**4-
     &                              T(I,J,KE-1)**4)
  700             CONTINUE
  900          CONTINUE
            ENDIF
         ENDIF
  500 CONTINUE

      RETURN
      END

*************************************************************************
*************************************************************************
      SUBROUTINE SOLCON
*************************************************************************
*   THIS SUBROUTINE RESETS THE CONDUCTIVITY OF THE SOLID. IN CALVIS     *
*   THE VISCOSITY ARE CALCULATED  AT EVERY CELL INCLUDING THOSE         *
*   CONTAINING SOLID ONES.                                              *
*************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL16/U0,UGRT,BUOY,CP0,PRT,COND0,VIS0,RHO0,
     &            TA,DTEMP,TWRITE,TTAPE,TMAX,GC,RAIR,NT
      COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
     &            JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

      DO 402 N=1,NCHIP
         IB=ICHPB(N)
         IE=IB+NCHPI(N)-1
         JB=JCHPB(N)
         JE=JB+NCHPJ(N)-1
         KB=KCHPB(N)
         KE=KB+NCHPK(N)-1
         DO 405 I=IB,IE-1
         DO 405 J=JB,JE-1
         DO 405 K=KB,KE-1
            COND(I,J,K)=COND0*CONS(N)
            CPM(I,J,K)=CPS(N)
            NOD(I,J,K)=1

C *** SET VALUE AT CORNER OR BOUNDARIES FOR BOUNDARY CONDITIONS
            IF (I.EQ.2) THEN
               COND(1,J,K)=COND(2,J,K)
               CPM (1,J,K)=CPM (2,J,K)
            ELSEIF (I.EQ.NI+3) THEN
```

```
               COND(NI+4,J,K)=COND(NI+3,J,K)
               CPM (NI+4,J,K)=CPM (NI+3,J,K)
            ENDIF

            IF (J.EQ.2) THEN
               COND(I,1,K)=COND(I,2,K)
               CPM (I,1,K)=CPM (I,2,K)
            ELSEIF (J.EQ.NJ+3) THEN
               COND(I,NJ+4,K)=COND(I,NJ+3,K)
               CPM (I,NJ+4,K)=CPM (I,NJ+3,K)
            ENDIF

            IF (K.EQ.2) THEN
               COND(I,J,1)=COND(I,J,2)
               CPM (I,J,1)=CPM (I,J,2)
            ELSEIF (K.EQ.NK+3) THEN
               COND(I,J,NK+4)=COND(I,J,NK+3)
               CPM (I,J,NK+4)=CPM (I,J,NK+3)
            ENDIF

  405    CONTINUE
  402 CONTINUE
       RETURN
       END

*******************************************************************************
*******************************************************************************
      SUBROUTINE STRESS
*******************************************************************************
*THIS SUBROUTINE CALCULATES THE SHEAR STRESSES

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL20/SIG11(25,25,15),SIG12(25,25,15),SIG22(25,25,15),
     &            SIG13(25,25,15),SIG23(25,25,15),SIG33(25,25,15)
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)

      DO 100 K=2,NK+3
      DO 100 J=2,NJ+3
      DO 100 I=2,NI+3

C *** CENTRAL LENGTH OF THE SCALAR CONTROL VOLUME
         DXP1=DXXC(I+1)
         DXI =DXXC(I)
         DXM1=DXXC(I-1)

         DYP1=DYYC(J+1)
         DYJ =DYYC(J)
         DYM1=DYYC(J-1)

         DZP1=DZZC(K+1)
```

```
          DZK =DZZC(K)
          DZM1=DZZC(K-1)

C *** SURFACE LENGTH OF THE CONTROL VOLUME
          DXN=DXXC(I)
          DXS=DXXC(I)
          DXF=DXXC(I)
          DXB=DXXC(I)

          DYF=DYYC(J)
          DYB=DYYC(J)
          DYE=DYYC(J)
          DYW=DYYC(J)

          DZE=DZZC(K)
          DZW=DZZC(K)
          DZN=DZZC(K)
          DZS=DZZC(K)

C *** CENTRAL LENGTH OF THE STAGGERED CONTROL VOLUME FOR TEMPERATURE
          DXEE=DXXS(I+2)
          DXE =DXXS(I+1)
          DXW =DXXS(I)
          DXWW=DXXS(I-1)

          DYNN=DYYS(J+2)
          DYN =DYYS(J+1)
          DYS =DYYS(J)
          DYSS=DYYS(J-1)

          DZFF=DZZS(K+2)
          DZF =DZZS(K+1)
          DZB =DZZS(K)
          DZBB=DZZS(K-1)

C *** CALCULATE THE AVERAGE VELOCITY IN THE CENTER OF THE CVOLUME
          UBAR=(U(I+1,J  ,K  )+U(I,J,K))/2.
          VBAR=(V(I  ,J+1,K  )+V(I,J,K))/2.
          WBAR=(W(I  ,J  ,K+1)+W(I,J,K))/2.

C *** CROSS-SECTIONAL AREA OF THE CV AT IT'S CENTER
          DXY=DXI*DYJ
          DYZ=DYJ*DZK
          DZX=DZK*DXI

C *** THE NOMRAL STRESSES
          SIG11(I,J,K)=2.*VIS(I,J,K)*((U(I+1,J,K)-U(I,J,K))/DXI+
     &                 VBAR*(DXN-DXS)/DXY+WBAR*(DXF-DXB)/DZX)
          SIG22(I,J,K)=2.*VIS(I,J,K)*((V(I,J+1,K)-V(I,J,K))/DYJ+
     &                 WBAR*(DYF-DYB)/DYZ+UBAR*(DYE-DYW)/DXY)
          SIG33(I,J,K)=2.*VIS(I,J,K)*((W(I,J,K+1)-W(I,J,K))/DZK+
     &                 UBAR*(DZE-DZW)/DZX+VBAR*(DZN-DZS)/DYZ)
 100  CONTINUE

C *** FOLLOWING DX, DY, DZ, ARE BASED ON THE LOCAL CV FOR SIG12
      DO 200 K=2,NK+4
```

```
      DO 200 J=2,NJ+4
      DO 200 I=2,NI+4

C *** CALCULATE THE LENGTH AT VARIOUS POSITIONS
         DXN=DXXS(I)
         DXS=DXXS(I)
         DXI=DXXS(I)

         DXE=DXXC(I)
         DXW=DXXC(I-1)

         DYE=DYYS(J)
         DYW=DYYS(J)
         DYJ=DYYS(J)

         DYN=DYYC(J)
         DYS=DYYC(J-1)

C *** THE AVERAGE VELOCITY IN THE CONTROL VOLUME
         UBAR=SILIN(U(I,J,K),U(I  ,J-1,K),DYN,DYS)
         VBAR=SILIN(V(I,J,K),V(I-1,J  ,K),DXE,DXW)

C *** AVERAGE VISCOSITY
         VIS12=BILIN(VIS(I  ,J,K),VIS(I  ,J-1,K),DYN,DYS,
     &               VIS(I-1,J,K),VIS(I-1,J-1,K),DYN,DYS,DXE,DXW)

C *** SHEAR STRESS SIG12
         SIGA=((V(I,J,K)-V(I-1,J  ,K))-VBAR*(DYE-DYW)/DYJ)/DXI
         SIGB=((U(I,J,K)-U(I  ,J-1,K))-UBAR*(DXN-DXS)/DXI)/DYJ
         SIG12(I,J,K)=VIS12*(SIGA+SIGB)

C *** FOLLOWING DX, DY, DZ, ARE BASED ON THE LOCAL CV FOR SIG13

C *** CALCULATE THE LENGTH AT VARIOUS POSITIONS
         DXF=DXXS(I)
         DXB=DXXS(I)
         DXI=DXXS(I)

         DXE=DXXC(I)
         DXW=DXXC(I-1)

         DZE=DZZS(K)
         DZW=DZZS(K)
         DZK=DZZS(K)

         DZF=DZZC(K)
         DZB=DZZC(K-1)

C *** THE AVERAGE VELOCITY IN THE CONTROL VOLUME
         UBAR=SILIN(U(I,J,K),U(I  ,J,K-1),DZF,DZB)
         WBAR=SILIN(W(I,J,K),W(I-1,J,K  ),DXE,DXW)

C *** AVERAGE VISCOSITY
         VIS13=BILIN(VIS(I  ,J,K),VIS(I  ,J,K-1),DZF,DZB,
     &               VIS(I-1,J,K),VIS(I-1,J,K-1),DZF,DZB,DXE,DXW)
```

```
C *** SHEAR STRESS SIG13
         SIGA=((W(I,J,K)-W(I-1,J,K  ))-WBAR*(DZE-DZW)/DZK)/DXI
         SIGB=((U(I,J,K)-U(I  ,J,K-1))-UBAR*(DXF-DXB)/DXI)/DZK
         SIG13(I,J,K)=VIS13*(SIGA+SIGB)

C *** FOLLOWING DX, DY, DZ, ARE BASED ON THE LOCAL CV FOR SIG23

C *** LENGTH AT VARIOUS POSITIONS
         DYF=DYYS(J)
         DYB=DYYS(J)
         DYJ=DYYS(J)

         DYN=DYYC(J)
         DYS=DYYC(J-1)

         DZN=DZZS(K)
         DZS=DZZS(K)
         DZK=DZZS(K)

         DZF=DZZC(K)
         DZB=DZZC(K-1)

C *** THE AVERAGE VELOCITY IN THE CONTROL VOLUME
         WBAR=SILIN(W(I,J,K),W(I,J-1,K  ),DYN,DYS)
         VBAR=SILIN(V(I,J,K),V(I,J  ,K-1),DZF,DZB)

C *** AVERAGE VISCOSITY
         VIS23=BILIN(VIS(I,J,K  ),VIS(I,J-1,K  ),DYN,DYS,
     &               VIS(I,J,K-1),VIS(I,J-1,K-1),DYN,DYS,DZF,DZB)

         SIGA=((V(I,J,K)-V(I,J  ,K-1))-VBAR*(DYF-DYB)/DYJ)/DZK
         SIGB=((W(I,J,K)-W(I,J-1,K  ))-WBAR*(DZN-DZS)/DZK)/DYJ
         SIG23(I,J,K)=VIS23*(SIGA+SIGB)

  200 CONTINUE
      RETURN
      END

*******************************************************************************
*******************************************************************************
      SUBROUTINE TCP
*******************************************************************************
*THIS SUBROUTINE CALCULATES THE NONDIMENSIONAL TEMPERATURE AT THE
*THERMOCOUPLE POSITIONS.
*******************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL32/T(25,25,15),R(25,25,15),P(25,25,15),C(25,25,15),
     &            U(25,25,15),V(25,25,15),W(25,25,15)
      COMMON/BL38/TCOUP(30),CX(30),CY(30),CZ(30),NTH(30,3),NTHCO

C *** CALCULATE SIZE OF CONTROL VOLUME CONTAINING THE THERMOCOUPLES
      DO 5100 N=1,NTHCO
```

```
          VOL=ABS((XC(NTH(N,1)+1)-XC(NTH(N,1)))*(YC(NTH(N,2)+1)-
     &         YC(NTH(N,2)))*(ZC(NTH(N,3)+1)-ZC(NTH(N,3))))
          TCOUP(N)=0.

          DO 5101 I=NTH(N,1),NTH(N,1)+1
             II=2*NTH(N,1)+1-I
             DO 5102 J=NTH(N,2),NTH(N,2)+1
                JJ=2*NTH(N,2)+1-J
                DO 5103 K=NTH(N,3),NTH(N,3)+1
                   KK=2*NTH(N,3)+1-K

C *** CORRECT TEMPERATURES FOR THERMOCOUPLES NOT LOCATED ON NODES
                   TVOL=ABS((XC(I)-CX(N))*(YC(J)-CY(N))*(ZC(K)-CZ(N)))
                   WVOL=TVOL/VOL
                   TCOUP(N)=TCOUP(N)+WVOL*T(II,JJ,KK)
 5103           CONTINUE
 5102        CONTINUE
 5101     CONTINUE
          IF (TCOUP(N).LT.TCOOL) TCOUP(N)=TCOOL
 5100  CONTINUE

       RETURN
       END

*************************************************************************
*************************************************************************
       SUBROUTINE TRID(IST,JST,KST,ISP,JSP,KSP,PHI)
*************************************************************************

       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
       COMMON/BL36/AP(25,25,15),AE(25,25,15),AW(25,25,15),AN(25,25,15),
      &            AS(25,25,15),AF(25,25,15),AB(25,25,15),SP(25,25,15),
      &            SU(25,25,15),RI(25,25,15)
       DIMENSION A(99),B(99),C(99),PHI(25,25,15)

C *** FORWARD SWEEP IN THE X DIRECTION (FROM IST TO ISP)
       A(IST-1)=0.
       C(IST-1)=0.
       DO 100 J=JST,JSP
       DO 100 K=KST,KSP
          DO 101 I=IST,ISP
             A(I)=AE(I,J,K)
             B(I)=AW(I,J,K)
             C(I)=AN(I,J,K)*PHI(I,J+1,K)+AS(I,J,K)*PHI(I,J-1,K)+
     &            AF(I,J,K)*PHI(I,J,K+1)+AB(I,J,K)*PHI(I,J,K-1)+SU(I,J,K)
             TERM=1./(AP(I,J,K)-B(I)*A(I-1))
             IF (ABS(A(I)).LE.1.0E-38) A(I)=0.0
             IF (ABS(B(I)).LE.1.0E-38) B(I)=0.0
             IF (ABS(C(I)).LE.1.0E-38) C(I)=0.0
             IF (ABS(TERM).LE.1.0E-38) TERM=0.0
             A(I)=A(I)*TERM
             C(I)=(C(I)+B(I)*C(I-1))*TERM
  101     CONTINUE
          PHI(ISP,J,K)=C(ISP)
          DO 102 I=ISP-1,IST,-1
```

```
            PHI(I,J,K)=A(I)*PHI(I+1,J,K)+C(I)
  102    CONTINUE
  100 CONTINUE

C *** FORWARD SWEEP IN THE Y DIRECTION (FROM JST TO JSP)
       A(JST-1)=0.
       C(JST-1)=0.
       DO 200 K=KST,KSP
       DO 200 I=IST,ISP
          DO 201 J=JST,JSP
             A(J)=AN(I,J,K)
             B(J)=AS(I,J,K)
             C(J)=AE(I,J,K)*PHI(I+1,J,K)+AW(I,J,K)*PHI(I-1,J,K)+
     &            AF(I,J,K)*PHI(I,J,K+1)+AB(I,J,K)*PHI(I,J,K-1)+SU(I,J,K)
             TERM=1./(AP(I,J,K)-B(J)*A(J-1))
             IF (ABS(A(J)).LE.1.0E-38) A(J)=0.0
             IF (ABS(B(J)).LE.1.0E-38) B(J)=0.0
             IF (ABS(C(J)).LE.1.0E-38) C(J)=0.0
             IF (ABS(TERM).LE.1.0E-38) TERM=0.0
             A(J)=A(J)*TERM
             C(J)=(C(J)+B(J)*C(J-1))*TERM
  201    CONTINUE
         PHI(I,JSP,K)=C(JSP)
         DO 202 J=JSP-1,JST,-1
            PHI(I,J,K)=A(J)*PHI(I,J+1,K)+C(J)
  202    CONTINUE
  200 CONTINUE

C *** FORWARD SWEEP IN THE Z DIRECTION (FROM KST TO KSP)
       A(KST-1)=0.
       C(KST-1)=0.
       DO 300 I=IST,ISP
       DO 300 J=JST,JSP
          DO 301 K=KST,KSP
             A(K)=AF(I,J,K)
             B(K)=AB(I,J,K)
             C(K)=AE(I,J,K)*PHI(I+1,J,K)+AW(I,J,K)*PHI(I-1,J,K)+
     &            AN(I,J,K)*PHI(I,J+1,K)+AS(I,J,K)*PHI(I,J-1,K)+SU(I,J,K)
             TERM=1./(AP(I,J,K)-B(K)*A(K-1))
             IF (ABS(A(K)).LE.1.0E-38) A(K)=0.0
             IF (ABS(B(K)).LE.1.0E-38) B(K)=0.0
             IF (ABS(C(K)).LE.1.0E-38) C(K)=0.0
             IF (ABS(TERM).LE.1.0E-38) TERM=0.0
             A(K)=A(K)*TERM
             C(K)=(C(K)+B(K)*C(K-1))*TERM
  301    CONTINUE
         PHI(I,J,KSP)=C(KSP)
         DO 302 K=KSP-1,KST,-1
            PHI(I,J,K)=A(K)*PHI(I,J,K+1)+C(K)
  302    CONTINUE
  300 CONTINUE

C *** REVERSE SWEEP IN X DIRECTION (FROM ISP TO IST)
       B(KSP+1)=0.
       C(KSP+1)=0.
       DO 600 I=ISP,IST,-1
```

```
      DO 600 J=JSP,JST,-1
         DO 601 K=KSP,KST,-1
            A(K)=AF(I,J,K)
            B(K)=AB(I,J,K)
            C(K)=AE(I,J,K)*PHI(I+1,J,K)+AW(I,J,K)*PHI(I-1,J,K)+
     &           AN(I,J,K)*PHI(I,J+1,K)+AS(I,J,K)*PHI(I,J-1,K)+SU(I,J,K)
            TERM=1./(AP(I,J,K)-A(K)*B(K+1))
            B(K)=B(K)*TERM
            C(K)=(C(K)+A(K)*C(K+1))*TERM
            IF (ABS(A(K)).LE.1.0E-38) A(K)=0.0
            IF (ABS(B(K)).LE.1.0E-38) B(K)=0.0
            IF (ABS(C(K)).LE.1.0E-38) C(K)=0.0
  601    CONTINUE
         PHI(I,J,KST)=C(KST)
         DO 602 K=KST+1,KSP
            PHI(I,J,K)=B(K)*PHI(I,J,K-1)+C(K)
  602    CONTINUE
  600 CONTINUE

C *** REVERSE SWEEP IN THE Y DIRECTION (FROM JSP TO JST)
      B(JSP+1)=0.
      C(JSP+1)=0.
      DO 500 K=KSP,KST,-1
      DO 500 I=ISP,IST,-1
         DO 501 J=JSP,JST,-1
            A(J)=AN(I,J,K)
            B(J)=AS(I,J,K)
            C(J)=AE(I,J,K)*PHI(I+1,J,K)+AW(I,J,K)*PHI(I-1,J,K)+
     &           AF(I,J,K)*PHI(I,J,K+1)+AB(I,J,K)*PHI(I,J,K-1)+SU(I,J,K)
            TERM=1./(AP(I,J,K)-A(J)*B(J+1))
            B(J)=B(J)*TERM
            C(J)=(C(J)+A(J)*C(J+1))*TERM
            IF (ABS(A(J)).LE.1.0E-38) A(J)=0.0
            IF (ABS(B(J)).LE.1.0E-38) B(J)=0.0
            IF (ABS(C(J)).LE.1.0E-38) C(J)=0.0
  501    CONTINUE
         PHI(I,JST,K)=C(JST)
         DO 502 J=JST+1,JSP
            PHI(I,J,K)=B(J)*PHI(I,J-1,K)+C(J)
  502    CONTINUE
  500 CONTINUE

C *** REVERSE SWEEP IN THE Z DIRECTION (FROM KSP TO KST)
      B(ISP+1)=0.
      C(ISP+1)=0.
      DO 400 J=JSP,JST,-1
      DO 400 K=KSP,KST,-1
         DO 401 I=ISP,IST,-1
            A(I)=AE(I,J,K)
            B(I)=AW(I,J,K)
            C(I)=AN(I,J,K)*PHI(I,J+1,K)+AS(I,J,K)*PHI(I,J-1,K)+
     &           AF(I,J,K)*PHI(I,J,K+1)+AB(I,J,K)*PHI(I,J,K-1)+SU(I,J,K)
            TERM=1./(AP(I,J,K)-A(I)*B(I+1))
            B(I)=B(I)*TERM
            C(I)=(C(I)+A(I)*C(I+1))*TERM
            IF (ABS(A(I)).LE.1.0E-38) A(I)=0.0
```

```
            IF (ABS(B(I)).LE.1.0E-38) B(I)=0.0
            IF (ABS(C(I)).LE.1.0E-38) C(I)=0.0
  401     CONTINUE
          PHI(IST,J,K)=C(IST)
          DO 402 I=IST+1,ISP
            PHI(I,J,K)=B(I)*PHI(I-1,J,K)+C(I)
  402     CONTINUE
  400 CONTINUE

      RETURN
      END

*****************************************************************************
*****************************************************************************
      SUBROUTINE VIEW
*****************************************************************************
*  NTHS         = TOTAL NUMBER OF NODES CONTAINING HEAT SORCE
*  NFX,NFY,NFZ  = NUMBER OF NODES IN HEAT SOURCE PER DIRECTION
*  FX,FY,FZ     = STARTING COORDINATES OF HEAT SOURCE
*  DXS,DYS,DZS  = LENGTH IN EACH DIRECTION ON THE SOUTH SURFACE
*                 OF THE ENCLOSURE
*  SXS,SYS,SZS  = COORDINATES OF THE SURFACE ELEMENT (USED TO
*                 CALCULATE THE DISTANCE TO HEAT SOURCE)
*  VFHSW,VFHSE  = VIEW FACTOR FROM THE HEAT SOURCE TO WEST, EAST
*                 SURFACES OF THE ENCLOSURE
*  RSQW,RSQE    = SQUARE OF DISTANCE FROM THE HEAT SOURCE TO THE
*                 WEST, EAST SURFACE ELEMENT.
*
*                 NODES 2 THRU NI,2 THRU NJ AND 2 THRU NK ARE CONTAINED
*                 INSIDE THE WALL, RADIATION EFFECTS INVOLVE ONLY NODES
*                 3 THRU NIM1, 3 THRU NJM1, AND 3 THRU NKM1
*
* VFHSBW(N,M,I,J),
* VFHSBE(N,M,I,J) = VIEW FACTOR FROM FIRE NODE N TO ELEMENT (I,J) OF
*                   WEST, EAST SURFACES OF CV M.
*****************************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)
      COMMON/BL1/DX,DY,DZ,DTIME,TCOOL,PI,Q,QR
      COMMON/BL2/X,Y,H,TFLR,TWAL
      COMMON/BL7/NI,NJ,NK,KRUN,NBLOR,NWRP
      COMMON/BL22/CPS(20),CONS(20),WFAN(20),NCHIP,ICHPB(20),NCHPI(20),
     &            JCHPB(20),NCHPJ(20),KCHPB(20),NCHPK(20)
      COMMON/BL37/VIS(25,25,15),COND(25,25,15),RESORM(40),
     &            CPM(25,25,15),NHSZ(3,2),NOD(25,25,15)
      COMMON/BL40/VFHSW(5,25,15),VFHSE(5,25,15),VFHSS(5,15,25),
     &            VFHSN(5,15,25),VFHSB(5,25,25),VFHSF(5,25,25)
      COMMON/BL41/VFHSBW(5,8,34,34),VFHSBE(5,8,34,34),VFHSBS(5,8,34,34),
     &            VFHSBN(5,8,34,34),VFHSBB(5,8,34,34),VFHSBF(5,8,34,34)

      NTHS=NHSZ(3,2)-NHSZ(3,1)+1
      DO 500 N=1,NTHS
      SUMM=0.
      NFX=NHSZ(1,1)
```

```
      NFY=NHSZ(2,1)
      NFZ=NHSZ(3,1)-1+N
      FX =XS(NFX+1)
      FY =YS(NFY+1)
      FZ =ZS(NFZ+1)

C *** VIEW FACTOR FROM FIRE TO WEST & EAST SURFACES OF ENCLOSURE
      DO 100 J=3,NJ+2
      DO 100 K=3,NK+2
         DYW=DYYC(J)
         DYE=DYYC(J)
         DZW=DZZC(K)
         DZE=DZZC(K)

         SXW=XS(3)
         SXE=XS(NI+3)
         SYW=YC(J)
         SYE=YC(J)
         SZW=ZC(K)
         SZE=ZC(K)

         DYZW=DYW*DZW
         DYZE=DYE*DZE

         RSQW=(FX-SXW)**2+(FY-SYW)**2+(FZ-SZW)**2
         RSQE=(FX-SXE)**2+(FY-SYE)**2+(FZ-SZE)**2

         VFHSW(N,J,K)=SQRT((FX-SXW)**2/RSQW)*DYZW/(4.0*PI*RSQW)
         VFHSE(N,J,K)=SQRT((FX-SXE)**2/RSQE)*DYZE/(4.0*PI*RSQE)
         SUMM=SUMM+VFHSW(N,J,K)+VFHSE(N,J,K)
  100 CONTINUE

C *** VIEW FACTOR FROM FIRE TO NORTH & SOUTH SURFACES OF ENCLOSURE
      DO 200 I=3,NI+2
      DO 200 K=3,NK+2
         DXS=DXXC(I)
         DXN=DXXC(I)
         DZS=DZZC(K)
         DZN=DZZC(K)

         SXN=XC(I)
         SXS=XC(I)
         SYN=YS(NJ+3)
         SYS=YS(3)
         SZN=ZC(K)
         SZS=ZC(K)

         DZXS=DXS*DZS
         DZXN=DXN*DZN

         RSQS=(FX-SXS)**2+(FY-SYS)**2+(FZ-SZS)**2
         RSQN=(FX-SXN)**2+(FY-SYN)**2+(FZ-SZN)**2

         VFHSS(N,K,I)=SQRT((FY-SYS)**2/RSQS)*DZXS/(4.0*PI*RSQS)
         VFHSN(N,K,I)=SQRT((FY-SYN)**2/RSQN)*DZXN/(4.0*PI*RSQN)
         SUMM=SUMM+VFHSS(N,K,I)+VFHSN(N,K,I)
```

```
  200 CONTINUE

C *** VIEW FACTOR FROM FIRE TO FRONT & BACK SURFACES OF ENCLOSURE
      DO 300 I=3,NI+2
      DO 300 J=3,NJ+2
         DXF=DXXC(I)
         DXB=DXXC(I)
         DYF=DYYC(J)
         DYB=DYYC(J)

         SXF=XC(I)
         SXB=XC(I)
         SYF=YC(J)
         SYB=YC(J)
         SZF=ZS(NK+3)
         SZB=ZS(3)

         DXYB=DXB*DYB
         DXYF=DXF*DYF

         RSQB=(FX-SXB)**2+(FY-SYB)**2+(FZ-SZB)**2
         RSQF=(FX-SXF)**2+(FY-SYF)**2+(FZ-SZF)**2

         VFHSB(N,I,J)=SQRT((FZ-SZB)**2/RSQB)*DXYB/(4.0*PI*RSQB)
         VFHSF(N,I,J)=SQRT((FZ-SZF)**2/RSQF)*DXYF/(4.0*PI*RSQF)
         SUMM=SUMM+VFHSB(N,I,J)+VFHSF(N,I,J)
  300 CONTINUE

C *** MODIFY VIEW FACTORS SO THEIR SUMMATION EQUALS UNITY.
      DO 150 J=3,NJ+2
      DO 150 K=3,NK+2
         VFHSW(N,J,K)=VFHSW(N,J,K)/SUMM
         VFHSE(N,J,K)=VFHSE(N,J,K)/SUMM
  150 CONTINUE
      DO 250 K=3,NK+2
      DO 250 I=3,NI+2
         VFHSS(N,K,I)=VFHSS(N,K,I)/SUMM
         VFHSN(N,K,I)=VFHSN(N,K,I)/SUMM
  250 CONTINUE
      DO 350 I=3,NI+2
      DO 350 J=3,NJ+2
         VFHSB(N,I,J)=VFHSB(N,I,J)/SUMM
         VFHSF(N,I,J)=VFHSF(N,I,J)/SUMM
  350 CONTINUE

      IF (NCHIP.LT.NBLOR) GOTO 500

C *** CALCULATE VIEW FACTORS FROM FIRE N TO INTERNAL SOLID BLOCKS
C *** (BLOCKS ARE ONLY THOSE NOT INCLUDED IN THE WALL)

C *** VIEW FACTOR FROM THE FIRE TO WEST & EAST SURFACES OF BLOCK M
      DO 900 M=1,NCHIP-NBLOR+1
         IB  =ICHPB(M+NBLOR-1)
         IE  =IB+NCHPI(M+NBLOR-1)-1
         JB  =JCHPB(M+NBLOR-1)
         JE  =JB+NCHPJ(M+NBLOR-1)-1
```

```
            KB  =KCHPB(M+NBLOR-1)
            KE  =KB+NCHPK(M+NBLOR-1)-1

            SXW=XS(IB)
            SXE=XS(IE)

            DO 400 JJ=JB,JE-1
            DO 400 KK=KB,KE-1
               DYW=DYYC(JJ)
               DYE=DYYC(JJ)
               DZW=DZZC(KK)
               DZE=DZZC(KK)

               SYW=YC(JJ)
               SYE=YC(JJ)
               SZW=ZC(KK)
               SZE=ZC(KK)

               DYZW=DYW*DZW
               DYZE=DYE*DZE

               RSQW=(FX-SXW)**2+(FY-SYW)**2+(FZ-SZW)**2
               RSQE=(FX-SXE)**2+(FY-SYE)**2+(FZ-SZE)**2
               VFHSBW(N,M,JJ,KK)=SQRT((FX-SXW)**2/RSQW)*DYZW/(4.0*PI*RSQW)
               VFHSBE(N,M,JJ,KK)=SQRT((FX-SXE)**2/RSQE)*DYZE/(4.0*PI*RSQE)

C *** MODIFY VIEW FACTORS DUE TO INTRODUCTION OF INTERNAL BLOCKS
               IF (SXE.LT.FX) VFHSBW(N,M,JJ,KK)=0.
               IF (SXW.GT.FX) VFHSBE(N,M,JJ,KK)=0.

C *** THE FIRE CAN'T SEE THE WEST AND EAST SURFACES OF THE BLOCK
               IF (SXW.LE.FX.AND.SXE.GE.FX) THEN
                  VFHSBW(N,M,JJ,KK)=0.
                  VFHSBE(N,M,JJ,KK)=0.
               ENDIF
  400       CONTINUE

C *** CHECK TO SEE IF ANY ELEMENT ON THE WALL IS SHADED BY A SOLID BLOCK.

C *** CHECK WEST AND EAST WALLS OF ENCLOSURE
            DO 410 J=3,NJ+2
            DO 410 K=3,NK+2

C *** THE BLOCK IS ON THE WEST SIDE OF THE FIRE
               IF (SXE.LT.FX) THEN
                  NVIW=NVIWX(FX,FY,FZ,XS(3),YC(J),ZC(K),SXE,
     &                       IB,JB,KB,IE,JE,KE)
                  IF (NVIW.EQ.1) VFHSW(N,J,K)=0.
               ENDIF

C *** THE BLOCK IS ON THE EAST SIDE OF THE FIRE
               IF (SXW.GT.FX) THEN
                  NVIW=NVIWX(FX,FY,FZ,XS(NI+3),YC(J),ZC(K),SXW,
     &                       IB,JB,KB,IE,JE,KE)
                  IF (NVIW.EQ.1) VFHSE(N,J,K)=0.
               ENDIF
```

```
  410    CONTINUE

C *** CHECK SOUTH AND NORTH WALLS OF THE ENCLOSURE
         DO 420 K=3,NK+2
         DO 420 I=3,NI+2

C *** THE BLOCK IS ON THE WEST SIDE OF THE FIRE
            IF (SXE.LT.FX.AND.XC(I).LT.FX) THEN
               NVIW=NVIWX(FX,FY,FZ,XC(I),YS(3),ZC(K),SXE,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSS(N,K,I)=0.
               NVIW=NVIWX(FX,FY,FZ,XC(I),YS(NJ+3),ZC(K),SXE,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSN(N,K,I)=0.
            ENDIF

C *** THE BLOCK IS ON THE EAST SIDE OF THE FIRE
            IF (SXW.GT.FX.AND.XC(I).GT.FX) THEN
               NVIW=NVIWX(FX,FY,FZ,XC(I),YS(3),ZC(K),SXW,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSS(N,K,I)=0.
               NVIW=NVIWX(FX,FY,FZ,XC(I),YS(NJ+3),ZC(K),SXW,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSN(N,K,I)=0.
            ENDIF
  420    CONTINUE

C *** CHECK BACK AND FRONT WALLS OF ENCLOSURE
         DO 430 I=3,NI+2
         DO 430 J=3,NJ+2

C *** THE BLOCK IS ON THE WEST SIDE OF THE FIRE
            IF (SXE.LT.FX.AND.XC(I).LT.FX) THEN
               NVIW=NVIWX(FX,FY,FZ,XC(I),YC(J),ZS(3),SXE,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSB(N,I,J)=0.
               NVIW=NVIWX(FX,FY,FZ,XC(I),YC(J),ZS(NK+3),SXE,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSF(N,I,J)=0.
            ENDIF

C *** THE BLOCK IS ON THE EAST SIDE OF THE FIRE
            IF (SXW.GT.FX.AND.XC(I).GT.FX) THEN
               NVIW=NVIWX(FX,FY,FZ,XC(I),YC(J),ZS(3),SXW,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSB(N,I,J)=0.
               NVIW=NVIWX(FX,FY,FZ,XC(I),YC(J),ZS(NK+3),SXW,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSF(N,I,J)=0.
            ENDIF
  430    CONTINUE

C *** VIEW FACTOR FROM FIRE TO NORTH & SOUTH SURFACES OF BLOCK M
         DO 600 II=IB,IE-1
         DO 600 KK=KB,KE-1
            DXN=DXXC(II)
```

```
            DXS=DXXC(II)
            DZN=DZZC(KK)
            DZS=DZZC(KK)

            SXN=XC(II)
            SXS=XC(II)
            SYN=YS(JE)
            SYS=YS(JB)
            SZN=ZC(KK)
            SZS=ZC(KK)

            DZXN=DXN*DZN
            DZXS=DXS*DZS

            RSQS=(FX-SXS)**2+(FY-SYS)**2+(FZ-SZS)**2
            RSQN=(FX-SXN)**2+(FY-SYN)**2+(FZ-SZN)**2
            VFHSBS(N,M,KK,II)=SQRT((FY-SYS)**2/RSQS)*DZXS/(4.0*PI*RSQS)
            VFHSBN(N,M,KK,II)=SQRT((FY-SYN)**2/RSQN)*DZXN/(4.0*PI*RSQN)

C *** MODIFY VIEW FACTORS DUE TO INTRODUCTION OF INTERNAL BLOCKS
            IF (SYN.LT.FY) VFHSBS(N,M,KK,II)=0.
            IF (SYS.GT.FY) VFHSBN(N,M,KK,II)=0.
            IF (SYS.LE.FY.AND.SYN.GE.FY) THEN
               VFHSBS(N,M,KK,II)=0.
               VFHSBN(N,M,KK,II)=0.
            ENDIF
  600    CONTINUE

C *** CHECK IF ANY ELEMENT ON WALL IS SHADED BY INTERNAL SOLID BLOCKS

C *** CHECK WEST AND EAST WALLS OF THE ENCLOSURE
         DO 610 J=3,NJ+2
         DO 610 K=3,NK+2

C *** THE BLOCK IS ON THE SOUTH SIDE OF THE FIRE
            IF (SYN.LT.FY.AND.YC(J).LT.FY) THEN
               NVIW=NVIWY(FX,FY,FZ,XS(3),YC(J),ZC(K),SYN,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSW(N,J,K)=0.
               NVIW=NVIWY(FX,FY,FZ,XS(NI+3),YC(J),ZC(K),SYN,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSE(N,J,K)=0.
            ENDIF

C *** THE BLOCK IS ON THE NORTH SIDE OF THE FIRE
            IF (SYS.GT.FY.AND.YC(J).GT.FY) THEN
               NVIW=NVIWY(FX,FY,FZ,XS(3),YC(J),ZC(K),SYS,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSW(N,J,K)=0.
               NVIW=NVIWY(FX,FY,FZ,XS(NI+3),YC(J),ZC(K),SYS,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSE(N,J,K)=0.
            ENDIF
  610    CONTINUE

C *** CHECK THE SOUTH AND NORTH WALLS OF THE ENCLOSURE
```

```
         DO 620 K=3,NK+2
         DO 620 I=3,NI+2

C *** THE BLOCK IS ON THE SOUTH SIDE OF THE FIRE
            IF (SYN.LT.FY) THEN
               NVIW=NVIWY(FX,FY,FZ,XC(I),YS(3),ZC(K),SYN,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSS(N,K,I)=0.
            ENDIF

C *** THE BLOCK IS ON THE NORTH SIDE OF THE FIRE
            IF (SYS.GT.FY) THEN
               NVIW=NVIWY(FX,FY,FZ,XC(I),YS(NJ+3),ZC(K),SYS,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSN(N,K,I)=0.
            ENDIF
  620    CONTINUE

C *** THE BACK AND FRONT WALLS OF THE ENCLOSURE
         DO 630 I=3,NI+2
         DO 630 J=3,NJ+2

C *** THE BLOCK IS ON THE SOUTH SIDE OF THE FIRE
            IF (SYN.LT.FY.AND.YC(J).LT.FY) THEN
               NVIW=NVIWY(FX,FY,FZ,XC(I),YC(J),ZS(3),SYN,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSB(N,I,J)=0.
               NVIW=NVIWY(FX,FY,FZ,XC(I),YC(J),ZS(NK+3),SYN,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSF(N,I,J)=0.
            ENDIF

C *** THE BLOCK IS ON THE NORTH SIDE OF THE FIRE
            IF (SYS.GT.FY.AND.YC(J).GT.FY) THEN
               NVIW=NVIWY(FX,FY,FZ,XC(I),YC(J),ZS(3),SYS,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSB(N,I,J)=0.
               NVIW=NVIWY(FX,FY,FZ,XC(I),YC(J),ZS(NK+3),SYS,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSF(N,I,J)=0.
            ENDIF
  630    CONTINUE

C *** CHECK VIEW FACTORS FROM FIRE TO BACK & FRONT SURFACES OF BLOCK M
         DO 700 II=IB,IE-1
         DO 700 JJ=JB,JE-1
            DXF=DXXC(II)
            DXB=DXXC(II)
            DYF=DYYC(JJ)
            DYB=DYYC(JJ)

            DXYB=DXB*DYB
            DXYF=DXF*DYF

            SXF=XC(II)
            SXB=XC(II)
```

```
            SYF=YC(JJ)
            SYB=YC(JJ)
            SZF=ZS(KE)
            SZB=ZS(KB)

            RSQB=(FX-SXB)**2+(FY-SYB)**2+(FZ-SZB)**2
            RSQF=(FX-SXF)**2+(FY-SYF)**2+(FZ-SZF)**2
            VFHSBB(N,M,II,JJ)=(FZ-SZB)**2*DXYB/(4.0*PI*RSQB**2)
            VFHSBF(N,M,II,JJ)=(FZ-SZF)**2*DXYF/(4.0*PI*RSQF**2)

C *** MODIFY VIEW FACTORS DUE TO INTRODUCTION OF INTERNAL SOLID BLOCKS
            IF (SZF.LT.FZ) VFHSBB(N,M,II,JJ)=0.
            IF (SZB.GT.FZ) VFHSBF(N,M,II,JJ)=0.
            IF (SZB.LE.FZ.AND.SZF.GE.FZ) THEN
               VFHSBB(N,M,II,JJ)=0.
               VFHSBF(N,M,II,JJ)=0.
            ENDIF
  700    CONTINUE

C *** CHECK IF ANY ELEMENT ON THE WALL IS SHADED BY SOLID BLOCK.

C *** THE WEST AND EAST WALLS OF THE ENCLOSURE
         DO 710 J=3,NJ+2
         DO 710 K=3,NK+2

C *** THE BLOCK IS ON THE BACK SIDE OF THE FIRE
            IF (SZF.LT.FZ.AND.ZC(K).LT.FZ) THEN
               NVIW=NVIWZ(FX,FY,FZ,XS(3),YC(J),ZC(K),SZF,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSW(N,J,K)=0.
               NVIW=NVIWZ(FX,FY,FZ,XS(NI+3),YC(J),ZC(K),SZF,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSE(N,J,K)=0.
            ENDIF

C *** THE BLOCK IS ON THE FRONT SIDE OF THE FIRE
            IF (SZB.GT.FZ.AND.ZC(K).GT.FZ) THEN
               NVIW=NVIWZ(FX,FY,FZ,XS(3),YC(J),ZC(K),SZB,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSW(N,J,K)=0.
               NVIW=NVIWZ(FX,FY,FZ,XS(NI+3),YC(J),ZC(K),SZB,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSE(N,J,K)=0.
            ENDIF
  710    CONTINUE

C *** CHECK THE SOUTH AND NORTH WALLS OF THE ENCLOSURE
         DO 720 K=3,NK+2
         DO 720 I=3,NI+2

C *** THE BLOCK IS ON THE BACK SIDE OF THE FIRE
            IF (SZF.LT.FZ.AND.ZC(K).LT.FZ) THEN
               NVIW=NVIWZ(FX,FY,FZ,XC(I),YS(3),ZC(K),SZF,
     &                    IB,JB,KB,IE,JE,KE)
               IF (NVIW.EQ.1) VFHSS(N,K,I)=0.
               NVIW=NVIWZ(FX,FY,FZ,XC(I),YS(NJ+3),ZC(K),SZF,
```

```
     &                    IB,JB,KB,IE,JE,KE)
                IF (NVIW.EQ.1) VFHSN(N,K,I)=0.
              ENDIF

C *** THE BLOCK IS ON THE FRONT SIDE OF THE FIRE
              IF (SZB.GT.FZ.AND.ZC(K).GT.FZ) THEN
                 NVIW=NVIWZ(FX,FY,FZ,XC(I),YS(3),ZC(K),SZB,
     &                    IB,JB,KB,IE,JE,KE)
                 IF (NVIW.EQ.1) VFHSS(N,K,I)=0.
                 NVIW=NVIWZ(FX,FY,FZ,XC(I),YS(NJ+3),ZC(K),SZB,
     &                    IB,JB,KB,IE,JE,KE)
                 IF (NVIW.EQ.1) VFHSN(N,K,I)=0.
              ENDIF
  720     CONTINUE

C *** CHECK THE BACK AND FRONT WALLS OF THE ENCLOSURE
          DO 730 I=3,NI+2
          DO 730 J=3,NJ+2

C *** THE BLOCK IS ON THE BACK SIDE OF THE FIRE
              IF (SZF.LT.FZ) THEN
                 NVIW=NVIWZ(FX,FY,FZ,XC(I),YC(J),ZS(3),SZF,
     &                    IB,JB,KB,IE,JE,KE)
                 IF (NVIW.EQ.1) VFHSB(N,I,J)=0.
              ENDIF

C *** THE BLOCK IS ON THE FRONT SIDE OF THE FIRE
              IF (SZB.GT.FZ) THEN
                 NVIW=NVIWZ(FX,FY,FZ,XC(I),YC(J),ZS(NK+3),SZB,
     &                    IB,JB,KB,IE,JE,KE)
                 IF (NVIW.EQ.1) VFHSF(N,I,J)=0.
              ENDIF
  730     CONTINUE
  900   CONTINUE
  500   CONTINUE

        RETURN
        END

*******************************************************************************
*******************************************************************************
        FUNCTION BILIN(V1,V2,D1,D2,V3,V4,D3,D4,D5,D6)
*******************************************************************************
* BI-LINEAR INTERPOLATION

        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        V12=(V1*D2+V2*D1)/(D1+D2)
        V34=(V3*D4+V4*D3)/(D3+D4)
        BILIN=(V12*D6+V34*D5)/(D5+D6)

        RETURN
        END

*******************************************************************************
*******************************************************************************
```

```
      INTEGER FUNCTION NVIWX(FX,FY,FZ,X1,Y1,Z1,X3,IB,JB,KB,IE,JE,KE)
*************************************************************************
*USED ONLY IN SUBROUTINE VIEW

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)

      NVIWX=0
      TPARA=(X3-X1)/(FX-X1)
      Y3=(FY-Y1)*TPARA+Y1
      Z3=(FZ-Z1)*TPARA+Z1
      IF (Y3.LE.YS(JE).AND.Y3.GE.YS(JB)) THEN
         IF (Z3.LE.ZS(KE).AND.Z3.GE.ZS(KB)) NVIWX=1
      ENDIF

      RETURN
      END

*************************************************************************
*************************************************************************
      INTEGER FUNCTION NVIWY(FX,FY,FZ,X1,Y1,Z1,Y3,IB,JB,KB,IE,JE,KE)
*************************************************************************
*USED ONLY IN SUBROUTINE VIEW

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)

      NVIWY=0
      TPARA=(Y3-Y1)/(FY-Y1)
      X3=(FX-X1)*TPARA+X1
      Z3=(FZ-Z1)*TPARA+Z1
      IF (X3.LE.XS(IE).AND.X3.GE.XS(IB)) THEN
         IF (Z3.LE.ZS(KE).AND.Z3.GE.ZS(KB)) NVIWY=1
      ENDIF

      RETURN
      END

*************************************************************************
*************************************************************************
      INTEGER FUNCTION NVIWZ(FX,FY,FZ,X1,Y1,Z1,Z3,IB,JB,KB,IE,JE,KE)
*************************************************************************
*USED ONLY IN SUBROUTINE VIEW

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/R4/XC(40),YC(40),ZC(40),XS(40),YS(40),ZS(40),DXXC(40),
     &          DYYC(40),DZZC(40),DXXS(40),DYYS(40),DZZS(40)

      NVIWZ=0
      TPARA=(Z3-Z1)/(FZ-Z1)
      Y3=(FY-Y1)*TPARA+Y1
      X3=(FX-X1)*TPARA+X1
      IF (Y3.LE.YS(JE).AND.Y3.GE.YS(JB)) THEN
         IF (X3.LE.XS(IE).AND.X3.GE.XS(IB)) NVIWZ=1
```

```
      ENDIF

      RETURN
      END

*******************************************************************************
*******************************************************************************
      FUNCTION SILIN(V1,V2,D1,D2)
*******************************************************************************
* SINGLE LINEAR INTERPOLATION

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      SILIN=(V1*D2+V2*D1)/(D1+D2)

      RETURN
      END
```

APPENDIX C. PROGRAM ISOTHERM

This program generates two dimensional isotherm plots using the National Center for Atmospheric Research (NCAR) Graphics software, from three dimensional temperature field data produced by program FIRE.

```
      PROGRAM ISOTHERM
C*************************************************************************
C                                                                        *
C  This program uses NCAR GRAPHICS to calculate and plot isotherms in    *
C  an arbitrary two dimensional section with uniform grid spacing        *
C  from a non-uniform three dimensional grid using linear interpola-     *
C  tion routines.  Isotherms are plotted using 14 equal size color       *
C  ranges based on the maximum temperature in the 3-D grid.              *
C                                                                        *
C*************************************************************************
C Variables used in this program. Arrays names are followed by left and
C    right parenthesis ().  (NOTE: some variables used in the plotting
C    routines have been specifically omitted from this list)
C
C  bth    :  Non-dimensional compartment breadth (X-direction)
C  cx     :  Interpolation parameter used to generate 2-D grid
C  cy     :  Interpolation parameter used to generate 2-D grid
C  c1     :  Interpolation parameter used to locate desired 3-D section
C  c2     :  Interpolation parameter used to locate desired 3-D section
C  c11    :  Interpolation parameter used to generate 2-D grid
C  c12    :  Interpolation parameter used to generate 2-D grid
C  c21    :  Interpolation parameter used to generate 2-D grid
C  c22    :  Interpolation parameter used to generate 2-D grid
C  dx     :  Non-dimensional 3-D grid size inside compartment, X dir.
C            (redefined in SUBROUTINE SCTN as non-dimensional uniform
C            grid spacing in the X direction for the 2-D plot)
C  dy     :  Non-dimensional 3-D grid size inside compartment, Y dir.
C            (redefined in SUBROUTINE SCTN as non-dimensional uniform
C            grid spacing in the Y direction for the 2-D plot)
C  dz     :  Non-dimensional 3-D grid size inside compartment, Z dir.
C  h      :  Non-dimensional Z length of compartment to model
C  iplot  :  Number of grids in X direction of 2-D grid
C  iscn   :  If >0 hold X constant, else =0
C  ixx    :  Number of X grid points in "xxs" array
C  jplot  :  Number of grids in Y direction of 2-D grid
C  jscn   :  If >0 hold Y constant, else =0
C  jyy    :  Number of Y grid points in "yys" array
C  kscn   :  If >0 hold Z constant, else =0
C  ncrmax :  Maximum number of contours
C  ni     :  Number of cells across compartment, X dir.
C  nj     :  Number of cells across compartment, Y dir.
C  nk     :  Number of cells across compartment, Z dir.
C  nstop  :  Control variable to terminate program
```

```
*  nview  :  Input value of desired plot (from SUBROUTINE SECTION)
*  t()    :  Imported array of temperature values
*  ta     :  Ambient absolute temperature prior to fire ignition
*  temp() :  Temperatures of desired section of 3-D grid
*  tflr   :  Non-dimensional thickness of compartment floor/ceiling
*  time   :  Dimensional time (in seconds) since fire ignition
*  tmax   :  Maximum temperature in "temp" array (redefined in SUBROUTINE
*            SECTION as maximum temperature in interpolated 2-D array "tt")
*  tt()   :  Section temperatures interpolated to 2-D grid
*  twal   :  Non-dimensional thickness of compartment walls
*  wth    :  Non-dimensional compartment width (Y-direction)
*  x      :  Non-dimensional X length of compartment to model
*  xfrn1  :  Starting 2-D, X coordinate of section to zoom in on
*  xfrn2  :  Ending 2-D, X coordinate of section to zoom in on
*  xs()   :  Non-dimensional grid locations
*  xscn   :  X location where Y-Z plane to be taken
*  xss()  :  Non-dimensional staggered grid locations
*  xtime  :  Non-dimensional time since fire ignition
*  xx     :  Non-dimensional length of 2-D X axis
*  xxs()  :  3-D grid points to be used as X coordinate in 2-D plot
*  y      :  Non-dimensional Y length of compartment to model
*  yfrn1  :  Starting 2-D, Y coordinate of section to zoom in on
*  yfrn2  :  Ending 2-D, Y coordinate of section to zoom in on
*  yscn   :  Y location where X-Z plane to be taken
*  yy     :  Non-dimensional length of 2-D Y axis
*  yys()  :  3-D grid points to be used as Y coordinate in 2-D plot
*  zscn   :  Z location where X-Y plane to be taken
*********************************************************************

c *** INPUT DATA FROM PROGRAM "FIRE"
      call INPUT

C *** GENERATE GRID
      call GRID

C *** DEFINE DESIRED SECTION
    5 call SECTION(nstop)

C *** CHECK IF FINISHED WITH PROGRAM
      if (nstop.ne.0) goto 9999

C *** LOCATE DESIRED SECTION TO PLOT
      call INTERPOL

C *** INTERPOLATE TO A 2-D PLOT
      call SCTN

C *** DEVELOP PLOT USING NCAR GRAPHICS
      call PLOT

C *** LOOP BACK FOR ANOTHER SECTION
      goto 5

C *** FINISHED WITH PROGRAM
 9999 end
*********************************************************************
```

```
      BLOCK DATA
*************************************************************************
*                                                                       *
*  The variables IPLOT and JPLOT must be changed in order to vary the   *
*  number of grids in the uniform 2-D plot.  NCRMAX must be changed to  *
*  vary the maximum number of colors allowed by the NCAR plots (NOTE:   *
*  NCRMAX must include one value for the default foreground color).     *
*                                                                       *
*************************************************************************

      implicit real*8 (a-h,o-z)
      common/ugrd/iplot,jplot,ncrmax
      common/sct1/iscn,jscn,kscn,nview

      data iplot,jplot,ncrmax/24,24,15/
      data nview/0/

      end
*************************************************************************
      SUBROUTINE SECTION(nstop)
*************************************************************************

      implicit real*8 (a-h,o-z)
      common/fctn/xfrn1,xfrn2,yfrn1,yfrn2
      common/sct1/iscn,jscn,kscn,nview
      common/sct2/xscn,yscn,zscn
      common/bl2/x,y,h,tflr,twal,ta

   99 print *
      print *
      print *, 'Enter   For:'
      print *, '_____   ____'
      print *, '  0     Finished with program'
      print *, '  1     Plan View (X-Y Plane)'
      print *, '  2     X-Z Profile'
      print *, '  3     Y-Z Profile'
      if(nview.le.0) goto 97
      print *, '  4     Enlargement of portion of previous plot'
   97 print *
      print *, 'Enter the desired isotherm plot:'
      read (*,*) nview
      nstop=0
      xfrn1=0.
      xfrn2=1.
      yfrn1=0.
      yfrn2=1.

c *** END PROGRAM
      if(nview.eq.0) then
         nstop=9999

C *** PLOT AN X-Y PLANE (a.k.a. Z section)
      elseif(nview.eq.1) then
         iscn=0
         jscn=0
         kscn=1
```

```
      xscn=0.
      yscn=0.
10    print *
7     format (1x,a/1x,a,f5.2,a)
      print 7,
     &   'Enter height of desired view plane in feet above the floor',
     &   '(Floor = 0.0; Ceiling = ',h,'):'
      read (*,*) zscn
      if(zscn.lt.0..or.zscn.gt.h) then
         print *
         print 5,'Input value must be between 0.0 and ',h
5        format (1x,a,f4.1)
         print *,'Try again.'
         go to 10
      endif

C *** PLOT AN X-Z PLANE (a.k.a. Y Section)
      elseif(nview.eq.2) then
         iscn=0
         jscn=1
         kscn=0
         xscn=0.
         zscn=0.
20    print *
      print 7,
     &        'Enter Y distance (in ft) of desired view plane from',
     &        'the left wall (Left Wall = 0.0; Right Wall = ',y,'):'
      read (*,*) yscn
      if(yscn.lt.0..or.yscn.gt.y) then
         print *
         print 5,'Input value must be between 0.0 and ',y
         print *,'Try again.'
         go to 20
      endif

C *** PLOT A Y-Z PLANE (a.k.a. X Section)
      elseif(nview.eq.3) then
         iscn=1
         jscn=0
         kscn=0
         yscn=0.
         zscn=0.
30    print *
      print 7,
     &        'Enter X distance (in ft) of desired view plane from ',
     &        'the left wall (Left Wall = 0.0; Right Wall = ',x,'):'
      read (*,*) xscn
      if(xscn.lt.0..or.xscn.gt.x) then
         print *
         print 5,'Input value must be between 0.0 and ',x
         print *,'Try again.'
         go to 30
      endif

c *** ZOOM IN ON A PORTION OF THE LAST PLOT
      elseif (nview.eq.4) then
```

```
   40    print *
         print *,'Enter the X & Y coordinates (expressed as a decimal'
         print *,'percentage of the whole plot, i.e. 25%=0.25) for the '
         print *,'lower left corner of the area to be enlarged.'
         read (*,*) xfrn1,yfrn1
         if(xfrn1.lt.0..or.xfrn1.gt.1..or.
     &      yfrn1.lt.0..or.yfrn1.gt.1.) then
            print *,'All coordinates must be between 0.00 and 1.00!'
            print *,'Please correct and reenter.'
            goto 40
         endif

   45    print *
         print *,'Enter the X & Y coordinates (expressed as a decimal'
         print *,'percentage of the whole plot, i.e. 25%=0.25) for the '
         print *,'upper right corner of the area to be enlarged.'
         read (*,*) xfrn2,yfrn2
         if(xfrn2.lt.0..or.xfrn2.gt.1..or.
     &      yfrn2.lt.0..or.yfrn2.gt.1.) then
            print *,'All coordinates must be between 0.00 and 1.00!'
            print *,'Please correct and reenter.'
            goto 45
         endif
         if(xfrn2.lt.xfrn1.or.yfrn2.lt.yfrn1) then
            print ('(1x,a,2(f4.2,a))'),
     &            'These coordinates must be greater than ',xfrn1,
     &            ' and ',yfrn1,' respectively.'
            print *,'Please correct and reenter.'
            goto 45
         endif

C *** ERROR ON ENTRY OF NVIEW
      else
         print *
         print *,'Incorrect response!  Please enter 0 - 4'
         goto 99
      endif

      return
      end
*****************************************************************************
      SUBROUTINE INPUT
*****************************************************************************
*                                                                           *
*  Input data from existing datafiles.                                      *
*                                                                           *
*****************************************************************************

      implicit real*8 (a-h,o-z)
      common/bl1/dx,dy,dz
      common/bl2/x,y,h,tflr,twal,ta
      common/bl7/ni,nj,nk
      common/data/t(25,25,15),temp(24,24),tt(24,24),xtime,ttmax

C *** READ IN DATA FROM EXISTING DATA FILE
      open(10,file='fire.dat',status='old')
```

```
      read(10,*) x,y,h,tflr,twal,ta
      read(10,*) ni,nj,nk
      rewind 10
      close (10)

      open (unit=9,file='plot.data',status='unknown')
      read(9,1001) xtime,t
 1001 format(4(f17.7))
      rewind 9
      close (9)

C *** DIMENSIONALIZE TEMPERATURES IN DEGRESS CELSIUS
      do 77 i=1,ni+4
      do 77 j=1,nj+4
      do 77 k=1,nk+4
         t(I,J,K)=ta*t(i,j,k)/1.8-273.16
   77 continue

      return
      end
*******************************************************************************
      SUBROUTINE GRID
*******************************************************************************

      implicit real*8 (a-h,o-z)
      common/r4/xss(40),yss(40),zss(40),xs(40),ys(40),zs(40)
      common/b11/dx,dy,dz
      common/b12/x,y,h,tflr,twal,ta
      common/b17/ni,nj,nk

C *** GENERATION OF THE GRIDS
      dx=x/(float(ni)*h)
      dy=y/(float(nj)*h)
      dz=h/(float(nk)*h)

C *** CALCULATE XSS,YSS,ZSS (COORDINATES OF STAGGERED CV'S)
      do 10 i=3,ni+3
         xss(i)=(i-3)*dx
   10 continue
      xss(2)=xss(3)-twal/(h*12.)
      xss(1)=xss(2)-twal/(h*12.)
      xss(ni+4)=xss(ni+3)+twal/(h*12.)
      xss(ni+5)=xss(ni+4)+twal/(h*12.)

      do 12 j=3,nj+3
         yss(j)=(j-3)*dy
   12 continue
      yss(2)=yss(3)-twal/(h*12.)
      yss(1)=yss(2)-twal/(h*12.)
      yss(nj+4)=yss(nj+3)+twal/(h*12.)
      yss(nj+5)=yss(nj+4)+twal/(h*12.)

      do 14 k=3,nk+3
         zss(k)=(k-3)*dz
   14 continue
      zss(2)=zss(3)-tflr/(h*12.)
```

```
      zss(1)=zss(2)-tflr/(h*12.)
      zss(nk+4)=zss(nk+3)+tflr/(h*12.)
      zss(nk+5)=zss(nk+4)+tflr/(h*12.)

C *** CONVERT TO CENTERED GRID FOR USE BY PROGRAM
      do 20 i=1,ni+4
         xs(i)=(xss(i)+xss(i+1))/2
   20 continue

      do 22 j=1,nj+4
         ys(j)=(yss(j)+yss(j+1))/2
   22 continue

      do 24 k=1,nk+4
         zs(k)=(zss(k)+zss(k+1))/2
   24 continue

      return
      end
*******************************************************************************
      SUBROUTINE INTERPOL
*******************************************************************************
*                                                                             *
*  This subroutine interpolates the 3-D section into the desired 2-D          *
*  section. It is a necessary step since any arbitrary section would          *
*  fall in between two grid points and must be linearly interpolated.         *
*                                                                             *
*******************************************************************************

      implicit real*8 (a-h,o-z)
      common/lmt1/ixx,jyy
      common/lmt2/xx,yy
      common/sct1/iscn,jscn,kscn,nview
      common/sct2/xscn,yscn,zscn
      common/data/t(25,25,15),temp(24,24),tt(24,24),xtime,ttmax
      common/r4/xss(40),yss(40),zss(40),xs(40),ys(40),zs(40)
      common/bl1/dx,dy,dz
      common/bl2/x,y,h,tflr,twal,ta
      common/bl7/ni,nj,nk
      common/ngs/xxs(40),yys(40)

C *** NONDIMENSIONALIZE NECESSARY VARIABLES
      if(nview.ne.4) then
         xscn = xscn / h
         yscn = yscn / h
         zscn = zscn / h
      endif
      wth = y / h
      bth = x / h

C *** LOCATE SECTION OF INTEREST

C *** X SECTION
      if (iscn .ne. 0) then
          do 10 i=2,ni+4
             if (xs(i) .ge. xscn .and. xs(i-1) .lt. xscn) then
```

```
                igi = i
             endif
  10      continue
          c1 = (xs(igi) - xscn) / (xs(igi) - xs(igi-1))
          c2 = 1. - c1

c *** INTERPOLATION DONE HERE
          do 20 j=1,nj+4
             do 20 k=1,nk+4
                temp(j,K) = c1 * t(igi-1,j,k) + c2 * t(igi,j,k)
  20      continue
          ixx = nj+4
          jyy = nk+4
          xx = wth
          yy = 1.
          do 22 i=1,ixx
             xxs(i) = ys(i)
  22      continue
          do 24 j=1,jyy
             yys(j) = zs(j)
  24      continue
       endif

C *** Y SECTION
       if (jscn .ne. 0) then
          do 30 j=2,nj+4
             if (ys(j) .ge. yscn .and. ys(j-1) .lt. yscn) then
                 jgi = j
             endif
  30      continue
          c1 = (ys(jgi) - yscn) / (ys(jgi) - ys(jgi-1))
          c2 = 1. - c1

c *** INTERPOLATION DONE HERE
          do 40 i=1,ni+4
             do 40 k=1,nk+4
                temp(i,k) = c1 * t(i,jgi-1,k) + c2 * t(i,jgi,k)
  40      continue
          ixx = ni+4
          jyy = nk+4
          xx = bth
          yy = 1.
          do 42 i=1,ixx
             xxs(i) = xs(i)
  42      continue
          do 44 j=1,jyy
             yys(j) = zs(j)
  44      continue
       endif

C *** Z SECTION
       if (kscn .ne. 0) then
          do 50 k=2,nk+4
             if (zs(k) .ge. zscn .and. zs(k-1) .lt. zscn) then
                 kgi = k
             endif
```

```
  50      continue
          c1 = (zs(kgi) - zscn) / (zs(kgi) - zs(kgi-1))
          c2 = 1. - c1

c *** INTERPOLATION DONE HERE
          do 60 i=1,ni+4
             do 60 j=1,nj+4
                temp(i,j) = c1 * t(i,j,kgi-1) + c2 * t(i,j,kgi)
  60      continue
          ixx = ni+4
          jyy = nj+4
          xx = bth
          yy = wth
          do 62 i=1,ixx
             xxs(i) = xs(i)
  62      continue
          do 64 j=1,jyy
             yys(j) = ys(j)
  64      continue
      endif

C *** DETERMINE MAXIMUM TEMPERATURE ON SECTION
      tmax = 0.
      do 70 i=1,ixx
         do 70 j=1,jyy
            if (temp(i,j) .gt. tmax) tmax = temp(i,j)
  70  continue
      print *,'THE MAXIMUM TEMPERATURE IS',tmax

      return
      end
*******************************************************************************
      SUBROUTINE SCTN
*******************************************************************************
*                                                                             *
*  In this subroutine the non-uniform grid is interpolated onto a             *
*  uniform grid for plotting purposes.  The routine has the option of         *
*  "blowing up" or "zooming in on" a certain portion for detailed             *
*  viewing.                                                                   *
*                                                                             *
*******************************************************************************

      implicit real*8 (a-h,o-z)
      common/data/t(25,25,15),temp(24,24),tt(24,24),xtime,ttmax
      common/lmt1/ixx,jyy
      common/lmt2/xx,yy
      common/r4/xss(40),yss(40),zss(40),xs(40),ys(40),zs(40)
      common/ngs/xxs(40),yys(40)
      common/ugrd/iplot,jplot,ncrmax
      common/fctn/xfrn1,xfrn2,yfrn1,yfrn2

c *** DETERMINE LIMITS OF INTERPOLATION
      xlmt1 = xfrn1 * xx
      xlmt2 = xfrn2 * xx
      ylmt1 = yfrn1 * yy
      ylmt2 = yfrn2 * yy
```

```
      dx = (xlmt2 - xlmt1) / float(iplot - 1)
      dy = (ylmt2 - ylmt1) / float(jplot - 1)

      do 20 i=1,iplot
         do 20 j=1,jplot

C *** LOCATE SECTION AND DEVELOP INTERPOLATION PARAMETERS
            xtmp = xlmt1 + dx * float(i - 1)
            ytmp = ylmt1 + dy * float(j - 1)
            do 10 ii=2,ixx
               if (xxs(ii) .gt. xtmp .and. xxs(ii-1) .lt. xtmp) then
                  iil = ii
               endif
   10       continue
            cx = (xxs(iil) - xtmp) / (xxs(iil) - xxs(iil-1))
            do 15 jj=2,jyy
               if (yys(jj) .gt. ytmp .and. yys(jj-1) .lt. ytmp) then
                  jjl = jj
               endif
   15       continue
            cy = (yys(jjl) - ytmp) / (yys(jjl) - yys(jjl-1))
            c11 = cx * cy
            c12 = cx * (1. - cy)
            c21 = (1. - cx) * cy
            c22 = (1. - cx) * (1. - cy)

C *** INTERPOLATION DONE HERE
            tt(i,j) = c11 * temp(iil-1,jjl-1) + c12 * temp(iil-1,jjl)
     &              + c21 * temp(iil,jjl-1) + c22 * temp(iil,jjl)
   20    continue

c *** DETERMINE MAXIMUM INTERPOLATED TEMPERATURE
      ttmax = 0.
      do 30 i=1,iplot
         do 30 j=1,jplot
            if (tt(i,j) .gt. ttmax) ttmax = tt(i,j)
   30 continue
      print *,'THE MAXIMUM INTERPOLATED TEMPERATURE IS',ttmax

      return
      end
*******************************************************************************
      SUBROUTINE PLOT
*******************************************************************************
*                                                                             *
*  This subroutine plots the isotherms using the CONRAN routine of            *
*  the NCAR Graphics package.                                                 *
*                                                                             *
*******************************************************************************

      common/lmt2/xx,yy
      common/data/t(25,25,15),temp(24,24),tt(24,24),xtime,ttmax
      common/ugrd/iplot,jplot,ncrmax
      common/sct1/iscn,jscn,kscn,nview
      common/sct2/xscn,yscn,zscn
      common/fctn/xfrn1,xfrn2,yfrn1,yfrn2
```

```
      common/bl2/x,y,h,tflr,twal,ta
      real*8 t,temp,tt,xtime,ttmax,xx,yy,xfrn1,xfrn2,yfrn1,yfrn2
      real*8 xscn,yscn,zscn,x,y,h,tflr,twal,ta

      dimension zdat(24,24),rwrk(1000),iwrk(1000),iama(120000)
      dimension iasf(13)
      dimension xcra(1000),ycra(1000)
      dimension iaia(10),igia(10)
      dimension lind(14)
      character*10 llbs(15)
      character*5 sec, elev
      character*45 title
      character*4 xl1,xl2,yl1,yl2

c *** DEFINE EXTERNAL SUBROUTINE TO SET COLORS
      external COLRAM

C *** SET PARAMETERS REQUIRED BY NCAR GRAPHICS ROUTINES
      data iasf/13*1/
      data lind/2,3,4,5,6,7,8,9,10,11,12,13,14,15/

***** CONVERT TO SINGLE PRECISION (NCAR WON'T WORK IN DOUBLE PRECISION) *****
C *** FIRE TIME WHEN DATA WAS TAKEN
      time=xtime*h/1.0

C *** SECTION TO PLOT
      if(iscn.gt.0) then
         scn=xscn*h
         xzoom1=xfrn1*y
         xzoom2=xfrn2*y
         yzoom1=yfrn1*h
         yzoom2=yfrn2*h
      elseif(jscn.gt.0) then
         scn=yscn*h
         xzoom1=xfrn1*x
         xzoom2=xfrn2*x
         yzoom1=yfrn1*h
         yzoom2=yfrn2*h
      else
         scn=zscn*h
         xzoom1=xfrn1*x
         xzoom2=xfrn2*x
         yzoom1=yfrn1*y
         yzoom2=yfrn2*y
      endif

C *** SET VERTICAL DIMENSION OF OUTPUT DISPLAY
c      if(iscn.gt.0.or.jscn.gt.0) then
c         ndim=14
c      else
         ndim=24
c      endif

c *** DETERMINE MAX AND MIN TEMPS FOR ENTIRE 3-D GRID
      cmax = -2.0e20
      cmin = 2.0e20
```

```
      do 10 i=1,24
         do 10 j=1,24
            do 10 k=1,14
               if(t(i,j,k).gt.cmax) cmax = t(i,j,k)
               if(t(i,j,k).lt.cmin) cmin = t(i,j,k)
  10  continue

      do 20 i=1,iplot
         do 20 j=1,jplot
            zdat(i,j) = tt(i,j)
  20  continue

c *** CALCULATE THE SPACINGS BETWEEN CONTOURS
      dc=(cmax-cmin)/real(ncrmax-1)

***** RUN NCAR GRAPHICS PACKAGE *****
C *** START NCAR GRAPHICS
      call GOPKS (6,0)
      call GOPWK (1,2,1)
      call GACWK (1)

C *** TURN OFF CLIPPING SO WORDS WILL PLOT
      call GSCLIP (0)
      call GSASF  (iasf)
      call GSFAIS (1)

C *** DEFINE COLORS
      call DFCLRS

C *** DEFINE VIEWPORT AND PLOT CONTOURS
      call CPSETR ('VPS - VIEWPORT SHAPE',0)
      if(kscn.gt.0) then
         call CPSETR ('VPB - VIEWPORT BOTTOM',.15)
         call CPSETR ('VPT - VIEWPORT TOP',.95)
      else
         call CPSETR ('VPB - VIEWPORT BOTTOM',.25)
         call CPSETR ('VPT - VIEWPORT TOP',.65)
      endif
      call CPSETR ('VPL - VIEWPORT LEFT',.1)
      call CPSETR ('VPR - VIEWPORT RIGHT',.9)
      call CPSETI ('NOF - NUMERIC OMISSION FLAGS',0)
      call CPSETI ('CLS - CONTOUR LEVEL SELECTOR',NCRMAX)
      call CPSETR ('CIS - CONTOUR INTERVAL SPECIFIER',dc)
      call CPSETI ('LLP - LINE LABEL POSITIONING',0)
      call CPSETR ('CMN - CONTOUR MINIMUM',cmin)
      call CPSETR ('CMX - CONTOUR MAXIMUM',cmax)
      call CPRECT (zdat,24,24,ndim,rwrk,1000,iwrk,1000)
      call ARINAM (iama,120000)
      call CPCLAM (zdat,rwrk,iwrk,iama)
      call ARSCAM (iama,xcra,ycra,1000,iaia,igia,10,COLRAM)
      call GSPLCI (0)
      call CPCLDR (zdat,rwrk,iwrk)
      call GSPLCI (1)

c *** CONVERT REAL VARIABLES TO CHARACTER VARIABLES FOR PLOTTING
C *** AND SET COLORS FOR PLOTTING
```

```
      call CPGETI ('NCL - NUMBER OF CONTOUR LEVELS',ncl)
      do 102 i=1,ncl
         call CPSETI ('PAI - PARAMETER ARRAY INDEX',i)
         call CPSETI ('AIA - AREA IDENTIFIER ABOVE',i)
         call CPSETI ('AIB - AREA IDENTIFIER BELOW',i-1)
         call CPGETR ('CLV - CONTOUR LEVEL VALUES',zlbs)
         call CPSETR ('ZDV - Z DATA VALUE',zlbs)
         call CPGETC ('ZDV - Z DATA VALUE',llbs(i))
  102 continue

      call CPSETR ('ZDV - Z DATA VALUE',time)
      call CPGETC ('ZDV - Z DATA VALUE',sec)

      call CPSETR ('ZDV - Z DATA VALUE',scn)
      call CPGETC ('ZDV - Z DATA VALUE',elev)

      call CPSETR ('ZDV - Z DATA VALUE',xzoom1)
      call CPGETC ('ZDV - Z DATA VALUE',xl1)

      call CPSETR ('ZDV - Z DATA VALUE',xzoom2)
      call CPGETC ('ZDV - Z DATA VALUE',xl2)

      call CPSETR ('ZDV - Z DATA VALUE',yzoom1)
      call CPGETC ('ZDV - Z DATA VALUE',yl1)

      call CPSETR ('ZDV - Z DATA VALUE',yzoom2)
      call CPGETC ('ZDV - Z DATA VALUE',yl2)

c *** CONSTRUCT LABEL BAR
      call LBSETI ('CBL - COLOR OF BOX LINES',0)
      if(kscn.gt.0) then
         call LBLBAR (0,.05,.95,.05,.10,14,1.,.5,lind,0,llbs,15,1)
      else
         call LBLBAR (0,.05,.95,.15,.20,14,1.,.5,lind,0,llbs,15,1)
      endif

***** LABEL AXIS AND TITLE PLOT *****
c *** X SECTION
      if(iscn.gt.0) then
         title='Y-Z PROFILE (X = '//elev//' FT.) AT '//sec//' SEC.'
         call PLCHHQ(3.,25.,title,.015,0.,-1.)
         call PLCHHQ(9.,-6.,'TEMPERATURE (CELSIUS)',.01,0.,-1.)
         call PLCHHQ(10.,0.,'BREADTH (Y-DIR)',.01,0.,-1.)
         call PLCHHQ(1.,0.,xl1,.01,0.,-1.)
         call PLCHHQ(23.,0.,xl2,.01,0.,-1.)
         call PLCHHQ(.5,11.0,'HEIGHT (Z-DIR)',.01,90.,0.)
         call PLCHHQ(.5,1.5,yl1,.01,90.,0.)
         call PLCHHQ(.5,23.,yl2,.01,90.,0.)

C *** Y SECTION
      elseif(jscn.gt.0) then
         title='X-Z PROFILE (Y = '//elev//' FT.) AT '//sec//' SEC.'
         call PLCHHQ(3.,25.,title,.015,0.,-1.)
         call PLCHHQ(9.,-6.,'TEMPERATURE (CELSIUS)',.01,0.,-1.)
         call PLCHHQ(10.,0.,'DEPTH (X-DIR)',.01,0.,-1.)
         call PLCHHQ(1.,0.,xl1,.01,0.,-1.)
```

```
         call PLCHHQ(23.,0.,x12,.01,0.,-1.)
         call PLCHHQ(.5,11.0,'HEIGHT (Z-DIR)',.01,90.,0.)
         call PLCHHQ(.5,1.5,y11,.01,90.,0.)
         call PLCHHQ(.5,23.,y12,.01,90.,0.)

C *** Z SECTION
      elseif(kscn.gt.0) then
         title='PLAN VIEW (Z = '//elev//' FT.) AT '//sec//' SEC.'
         call PLCHHQ(3.5,24.5,title,.015,0.,-1.)
         call PLCHHQ(8.,-2.5,'TEMPERATURE (CELSIUS)',.01,0.,-1.)
         call PLCHHQ(10.,.5,'DEPTH (X-DIR)',.01,0.,-1.)
         call PLCHHQ(1.,.5,x11,.01,0.,-1.)
         call PLCHHQ(23.,.5,x12,.01,0.,-1.)
         call PLCHHQ(.5,9.0,'BREADTH (Y-DIR)',.01,90.,-1.)
         call PLCHHQ(.5,1.5,y11,.01,90.,0.)
         call PLCHHQ(.5,23.5,y12,.01,90.,0.)
      endif

C *** DRAW BOUNDARY AROUND VIEWPORT
      call BNDARY

c *** ADVANCE FRAME FOR NEXT PLOT
      call FRAME

C *** FINISHED WITH NCAR GRAPHICS
      call GCLRWK (1,1)
      call GDAWK  (1)
      call GCLWK  (1)
      call GCLKS

      return
      end
*******************************************************************************
      SUBROUTINE DFCLRS
*******************************************************************************
*                                                                             *
*  Define colors using  Red-Green-Blue (RGB) triples.                         *
*                                                                             *
*******************************************************************************

      dimension rgbv(3,15)
      data rgbv/ 0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,
     &           0.00,0.00,1.00,0.00,0.50,0.75,0.00,0.75,0.50,
     &           0.00,1.00,0.00,0.65,1.00,0.00,1.00,1.00,0.00,
     &           1.00,0.75,0.00,1.00,0.50,0.00,1.00,0.00,0.00,
     &           1.00,0.10,0.40,1.00,0.40,0.70,1.00,0.70,1.00/

c *** DEFINE DEFAULT BACKGROUND COLOR AS WHITE
      call GSCR (1,0,1.,1.,1.)
c
C *** DEFINE REMAINING COLORS
C *** (NOTE: i=1 is default foreground color and is set to black)
      do 101 i=1,15
         call GSCR (1,i,rgbv(1,i),rgbv(2,i),rgbv(3,i))
  101 continue
```

```
      return
      end
*****************************************************************************
      SUBROUTINE COLRAM (xcra,ycra,ncra,iaia,igia,naia)
*****************************************************************************
*                                                                           *
*  This subroutine is used by NCAR GRAPHICS (call DEFCLRS) to assign        *
*  the colors to the contour levels.  It must be declared EXTERNAL          *
*  before any calls are made to any NCAR subroutines.                       *
*                                                                           *
*****************************************************************************

      dimension xcra(*),ycra(*),iaia(*),igia(*)
      ifll = 1

      do 101 i=1,naia
         if (iaia(i) .lt. 0) ifll = 0
  101 continue
      if (ifll .ne. 0) then
          ifll = 0
          do 102 i=1,naia
             if (igia(i) .eq. 3) ifll = iaia(i)
  102     continue
          if (ifll .gt. 0 .and. ifll .lt. 15) then
              call GSFACI (ifll)
              call GFA (ncra-1,xcra,ycra)
          endif
      endif

      return
      end
*****************************************************************************
      SUBROUTINE BNDARY
*****************************************************************************
*                                                                           *
*  This subroutine defines the edge of the plot frame and draws a line      *
*  at its location.                                                         *
*                                                                           *
*****************************************************************************

      call PLOTIF (0.,0.,0)
      call PLOTIF (1.,0.,1)
      call PLOTIF (1.,1.,1)
      call PLOTIF (0.,1.,1)
      call PLOTIF (0.,0.,1)
      call PLOTIF (0.,0.,2)

      return
      end
```

APPENDIX D. PROGRAM VELOCITY

This program plots two dimensional velocity profiles, using the National Center for Atmospheric Research (NCAR) Graphics software, from the three dimensional velocity field generated by the program FIRE.

```
      PROGRAM VELOCITY
************************************************************************
*                                                                      *
*  This program uses NCAR GRAPHICS to calculate and plot velocity      *
*  vectors in an arbitrary two dimensional section with uniform grid   *
*  spacing from a non-uniform three dimensional grid using linear      *
*  interpolation routines.                                             *
*                                                                      *
************************************************************************
*Variables:
*  nstop  :  Control variable to end program
*  xscn   :  X location where Y-Z plane to be taken
*  yscn   :  Y location where X-Z plane to be taken
*  zscn   :  Z location where X-Y plane to be taken
*  iscn   :  If >0 hold X constant, else =0
*  jscn   :  If >0 hold Y constant, else =0
*  kscn   :  If >0 hold Z constant, else =0
*  xfrn1  :  Starting 2-D, X coordinate of section to zoom in on
*  xfrn2  :  Ending 2-D, X coordinate of section to zoom in on
*  yfrn1  :  Starting 2-D, Y coordinate of section to zoom in on
*  yfrn2  :  Ending 2-D, Y coordinate of section to zoom in on
*  t()    :  Dummy variable necessary due to format of input datafile
*  u()    :  Non-dimensional input value of X component of velocity
*  v()    :  Non-dimensional input value of Y component of velocity
*  w()    :  Non-dimensional input value of Z component of velocity
*  iplot  :  Number of grids in X direction of 2-D grid
*  jplot  :  Number of grids in Y direction of 2-D grid
*  x      :  Non-dimensional X length of compartment to model
*  y      :  Non-dimensional Y length of compartment to model
*  h      :  Non-dimensional Z length of compartment to model
*  tflr   :  Non-dimensional thickness of compartment floor/ceiling
*  twal   :  Non-dimensional thickness of compartment walls
*  ta     :  Non-dimensional ambient temperature
*  dx     :  Non-dimensional 3-D grid size inside compartment, X dir.
*  dy     :  Non-dimensional 3-D grid size inside compartment, Y dir.
*  dz     :  Non-dimensional 3-D grid size inside compartment, Z dir.
*  ni     :  Number of cells across compartment, X dir.
*  nj     :  Number of cells across compartment, Y dir.
*  nk     :  Number of cells across compartment, Z dir.
*  nview  :  Input value of desired plot (from SUBROUTINE SECTION)
************************************************************************

C *** READ IN DATA GENERATED BY PROGRAM "FIRE"
```

```
      call INPUT

C *** GENERATE GRID
      call GRID

C *** DEFINE DESIRED SECTION
    5 call SECTION(nstop)

C *** CHECK IF FINISHED WITH PROGRAM
      if (nstop.gt.0) goto 9999

C *** LOCATE DESIRED SECTION
      call INTERPOL

C *** INTERPOLATE TO A 2-D PLOT
      call SCTN

C *** GENERATE VELOCITY VECTOR PLOT USING NCAR GRAPHICS
      call PLOT

C *** LOOP BACK FOR ANOTHER SECTION
      goto 5

C *** FINISHED WITH PROGRAM
 9999 end
*******************************************************************************
      BLOCK DATA
*******************************************************************************
*                                                                             *
*  The variables IPLOT and JPLOT define the number of uniformly spaced       *
*  grids to be used in generating the 2-D plot and must be changed if         *
*  a different grid spacing is desired.                                       *
*                                                                             *
*******************************************************************************

      implicit real*8 (a-h,o-z)
      common/ugrd/iplot,jplot
      common/sct1/iscn,jscn,kscn,nview

      data iplot,jplot/24,24/
      data nview/0/

      end
*******************************************************************************
      SUBROUTINE INPUT
*******************************************************************************
*                                                                             *
*  Input data from existing datafiles.                                        *
*                                                                             *
*******************************************************************************

      implicit real*8 (a-h,o-z)
      common/bl1/dx,dy,dz
      common/bl2/x,y,h,tflr,twal,ta,xtime
      common/bl7/ni,nj,nk
      common/data/t(25,25,15),u(25,25,15),v(25,25,15),w(25,25,15),
```

```
     &            uu(24,24),vv(24,24),uu1(24,24),vv1(24,24)

C *** READ IN DATA FROM EXISTING DATA FILE
      open(10,file='fire.dat',status='old')
      read(10,*) x,y,h,tflr,twal,ta
      read(10,*) ni,nj,nk
      rewind 10
      close (10)

      open (unit=9,file='plot.data',status='unknown')
      read(9,1000) xtime,t,u,v,w
 1000 format(4(f17.7))

C *** DIMENSIONALIZE VELOCITIES TO CM/SEC
      do 10 i=1,ni+4
         do 10 j=1,nj+4
            do 10 k=1,nk+4
               u(i,j,k)=u(i,j,k)*30.25
               v(i,j,k)=v(i,j,k)*30.25
               w(i,j,k)=w(i,j,k)*30.25
   10 continue

      return
      end
*************************************************************************
      SUBROUTINE GRID
*************************************************************************

      implicit real*8 (a-h,o-z)
      common/r4/xss(40),yss(40),zss(40),xs(40),ys(40),zs(40)
      common/bl1/dx,dy,dz
      common/bl2/x,y,h,tflr,twal,ta,xtime
      common/bl7/ni,nj,nk

C *** GENERATION OF THE GRIDS
      dx=x/(float(ni)*h)
      dy=y/(float(nj)*h)
      dz=h/(float(nk)*h)

C *** CALCULATE XSS,YSS,ZSS (COORDINATES OF STAGGERED CV'S)
      do 10 i=3,ni+3
         xss(i)=(i-3)*dx
   10 continue
      xss(2)=xss(3)-twal/(h*12.)
      xss(1)=xss(2)-twal/(h*12.)
      xss(ni+4)=xss(ni+3)+twal/(h*12.)
      xss(ni+5)=xss(ni+4)+twal/(h*12.)

      do 12 j=3,nj+3
         yss(j)=(j-3)*dy
   12 continue
      yss(2)=yss(3)-twal/(h*12.)
      yss(1)=yss(2)-twal/(h*12.)
      yss(nj+4)=yss(nj+3)+twal/(h*12.)
      yss(nj+5)=yss(nj+4)+twal/(h*12.)
```

```
      do 14 k=3,nk+3
         zss(k)=(k-3)*dz
   14 continue
      zss(2)=zss(3)-tflr/(h*12.)
      zss(1)=zss(2)-tflr/(h*12.)
      zss(nk+4)=zss(nk+3)+tflr/(h*12.)
      zss(nk+5)=zss(nk+4)+tflr/(h*12.)

C *** CONVERT TO CENTERED GRID FOR USE BY PROGRAM
      do 20 i=1,ni+4
         xs(i)=(xss(i)+xss(i+1))/2
   20 continue

      do 22 j=1,nj+4
         ys(j)=(yss(j)+yss(j+1))/2
   22 continue

      do 24 k=1,nk+4
         zs(k)=(zss(k)+zss(k+1))/2
   24 continue

      return
      end
*************************************************************************
      SUBROUTINE SECTION(nstop)
*************************************************************************

      implicit real*8 (a-h,o-z)
      common/fctn/xfrn1,xfrn2,yfrn1,yfrn2
      common/sct1/iscn,jscn,kscn,nview
      common/sct2/xscn,yscn,zscn
      common/bl2/x,y,h,tflr,twal,ta,xtime


   99 print *
      print *
      print *,'ENTER   FOR:'
      print *,'-----   ----'
      print *,'  0     Finished with program'
      print *,'  1     Plan View (X-Y plane)'
      print *,'  2     Elevation (X-Z plane)'
      print *,'  3     Elevation (Y-Z plane)'
      if(nview.le.0) goto 97
      print *,'  4     Enlargement of portion of previous plot'
   97 print *
      print *,'Enter your desired vector field plot:'
      read (*,*) nview
      nstop=0
      xfrn1=0.
      xfrn2=1.
      yfrn1=0.
      yfrn2=1.

C *** END PROGRAM
      if (nview.eq.0) then
         nstop=9999
```

```
C *** PLOT AN X-Y PLANE (aka Z SECTION)
      elseif (nview.eq.1) then
         iscn=0
         jscn=0
         kscn=1
         xscn=0.
         yscn=0.
   10    print *
    7    format (1x,a/1x,a,f5.2,a)
         print 7,
     &         'Enter desired height of desired plot above the floor',
     &         '(Floor = 0.0; Ceiling = ',h,')'
         read (*,*) zscn
         if(zscn.lt.0..or.zscn.gt.h) then
            print *
            print 5,'Input value must be between 0.0 and ',h
    5       format (1x,a,f4.1)
            print *,'Try again.'
            go to 10
      endif

C *** PLOT AN X-Z PLANE (aka Y SECTION)
      elseif (nview.eq.2) then
         iscn=0
         jscn=1
         kscn=0
         xscn=0.
         zscn=0.
   20    print *
         print 7,
     &    'Enter Y distance (in ft) of desired plot from the left wall',
     &         '(Left Wall = 0.0; Right Wall = ',y,')'
         read (*,*) yscn
         if(yscn.lt.0..or.yscn.gt.y) then
            print *
            print 5,'Input value must be between 0.0 and ',y
            print *,'Try again.'
            go to 20
         endif

C *** PLOT A Y-Z PLANE (aka X SECTION)
      elseif (nview.eq.3) then
         iscn=1
         jscn=0
         kscn=0
         zscn=0.
         yscn=0.
   30    print *
         print 7,
     &   'Enter X distance (in ft) of desired plot from the right wall',
     &         '(Left Wall = 0.0; Right Wall = ',x,')'
         read (*,*) xscn
         if(xscn.lt.0..or.xscn.gt.x) then
            print *
            print 5,'Input value must be between 0.0 and ',x
```

```
            print *,'Try again.'
            go to 30
         endif

C *** ZOOM IN ON A PORTION OF THE LAST PLOT
      elseif (nview.eq.4) then
   40    print *
         print *,'Enter the X & Y coordinates (expressed as a decimal'
         print *,'percentage of the whole plot, i.e. 25%=0.25) for the '
         print *,'lower left corner of the area to be enlarged.'
         read (*,*) xfrn1,yfrn1
         if(xfrn1.lt.0..or.xfrn1.gt.1..or.
     &      yfrn1.lt.0..or.yfrn1.gt.1.) then
            print *,'All coordinates must be between 0.00 and 1.00!'
            print *,'Please correct and reenter.'
            goto 40
         endif

   45    print *
         print *,'Enter the X & Y coordinates (expressed as a decimal'
         print *,'percentage of the whole plot, i.e. 25%=0.25) for the '
         print *,'upper right corner of the area to be enlarged.'
         read (*,*) xfrn2,yfrn2
         if(xfrn2.lt.0..or.xfrn2.gt.1..or.
     &      yfrn2.lt.0..or.yfrn2.gt.1.) then
            print *,'All coordinates must be between 0.00 and 1.00!'
            print *,'Please correct and reenter.'
            goto 45
         endif
         if(xfrn2.lt.xfrn1.or.yfrn2.lt.yfrn1) then
            print ('(1x,a,2(f4.2,a))'),
     &            'These coordinates must be greater than ',xfrn1,
     &            ' and ',yfrn1,' respectively.'
            print *,'Please correct and reenter.'
            goto 45
         endif

C *** ERROR ON ENTRY OF NVIEW
      else
         print *
         print *,'Incorrect response!  Please enter 0 - 4'
         goto 99
      endif

      return
      end

*******************************************************************************
      SUBROUTINE INTERPOL
*******************************************************************************
*                                                                             *
*  This subroutine interpolates to the desired section. It is                 *
*  necessary since any arbitrary section would fall in between two            *
*  grid points and must be linearly interpolated.                             *
*                                                                             *
*******************************************************************************
```

```
      implicit real*8 (a-h,o-z)
      common/data/t(25,25,15),u(25,25,15),v(25,25,15),w(25,25,15),
     &           uu(24,24),vv(24,24),uu1(24,24),vv1(24,24)
      common/lmt1/ixx,jyy
      common/lmt2/xx,yy
      common/sct1/iscn,jscn,kscn,nview
      common/ngs/xxs(40),yys(40),x1(40),y1(40)
      common/r4/xss(40),yss(40),zss(40),xs(40),ys(40),zs(40)
      common/bl1/dx,dy,dz
      common/sct2/xscn,yscn,zscn
      common/bl2/x,y,h,tflr,twal,ta,xtime
      common/bl7/ni,nj,nk

C *** NON-DIMENSIONALIZE REQUIRED VARIABLES
      if(nview.ne.4) then
         xscn = xscn / h
         yscn = yscn / h
         zscn = zscn / h
      endif
      wth = y / h
      bth = x / h

***** LOCATE SECTION OF INTEREST *****

C *** X SECTION
      if (iscn .ne. 0) then
          do 10 i=2,ni+4
             if (xs(i) .ge. xscn .and. xs(i-1) .lt. xscn) then
                 igi = i
             endif
  10      continue
          c1 = (xs(igi) - xscn) / (xs(igi) - xs(igi-1))
          c2 = 1. - c1

C *** INTERPOLATION DONE HERE
          do 20 j=2,nj+4
             do 20 k=2,nk+4
                vv(j,k) = c1 * w(igi-1,j,k) + c2 * w(igi,j,k)
                uu(j,k) = c1 * v(igi-1,j,k) + c2 * v(igi,j,k)
  20      continue
          ixx = nj+4
          jyy = nk+4
          xx = wth
          yy = 1.
          do 22 i=1,ixx
             xxs(i) = ys(i)
             x1(i) = yss(i)
  22      continue
          do 24 j=1,jyy
             yys(j) = zs(j)
             y1(j) = zss(j)
  24      continue
      endif

C *** Y SECTION
```

```
      if (jscn .ne. 0) then
          do 30 j=2,nj+4
             if (ys(j) .ge. yscn .and. ys(j-1) .lt. yscn) then
                 jgi = j
             endif
 30       continue
          c1 = (ys(jgi) - yscn) / (ys(jgi) - ys(jgi-1))
          c2 = 1. - c1

C *** INTERPOLATION DONE HERE
          do 40 i=2,ni+4
             do 40 k=2,nk+4
                uu(i,k) = c1 * u(i,jgi-1,k) + c2 * u(i,jgi,k)
                vv(i,k) = c1 * w(i,jgi-1,k) + c2 * w(i,jgi,k)
 40       continue
          ixx = ni+4
          jyy = nk+4
          xx = bth
          yy = 1.
          do 42 i=1,ixx
             xxs(i) = xs(i)
             x1(i) = xss(i)
 42       continue
          do 44 j=1,jyy
             yys(j) = zs(j)
             y1(j) = zss(j)
 44       continue
      endif

C *** Z SECTION
      if (kscn .ne. 0) then
          do 50 k=2,nk+4
             if (zs(k) .ge. zscn .and. zs(k-1) .lt. zscn) then
                 kgi = k
             endif
 50       continue
          c1 = (zs(kgi) - zscn) / (zs(kgi) - zs(kgi-1))
          c2 = 1. - c1

C *** INTERPOLATION DONE HERE
          do 60 i=2,ni+4
             do 60 j=2,nj+4
                uu(i,j) = c1 * u(i,j,kgi-1) + c2 * u(i,j,kgi)
                vv(i,j) = c1 * v(i,j,kgi-1) + c2 * v(i,j,kgi)
 60       continue
          ixx = ni+4
          jyy = nj+4
          xx = bth
          yy = wth
          do 62 i=1,ixx
             xxs(i) = xs(i)
             x1(i) = xss(i)
 62       continue
          do 64 j=1,jyy
             yys(j) = ys(j)
             y1(j) = yss(j)
```

```
  64      continue
      endif

C *** DETERMINE MAXIMUM VELOCITY COMPONENTS
      cx = 0.
      cy = 0.
      do 70 i=1,ixx
         do 70 j=1,jyy
            if (abs(uu(i,j)) .gt. cx) cx = abs(uu(i,j))
            if (abs(vv(i,j)) .gt. cy) cy = abs(vv(i,j))
  70  continue
      print *,'THE MAXIMUM X-DIRECTION VELOCITY IS',cx
      print *,'THE MAXIMUM Y-DIRECTION VELOCITY IS',cy

      return
      end
*******************************************************************************
      SUBROUTINE SCTN
*******************************************************************************
*                                                                             *
*                                                                             *
*  In this subroutine the non-uniform grid is interpolated onto a             *
*  uniform grid for plotting purposes.  The routine has the option of         *
*  "blowing up" or "zooming in on" a certain portion for detailed             *
*  viewing.                                                                   *
*                                                                             *
*******************************************************************************

      implicit real*8 (a-h,o-z)
      common/data/t(25,25,15),u(25,25,15),v(25,25,15),w(25,25,15),
     &            uu(24,24),vv(24,24),uu1(24,24),vv1(24,24)
      common/lmt1/ixx,jyy
      common/lmt2/xx,yy
      common/r4/xss(40),yss(40),zss(40),xs(40),ys(40),zs(40)
      common/ngs/xxs(40),yys(40),x1(40),y1(40)
      common/ugrd/iplot,jplot
      common/fctn/xfrn1,xfrn2,yfrn1,yfrn2

C *** DETERMINE LIMITS OF INTERPOLATION
      xlmt1 = xfrn1 * xx
      xlmt2 = xfrn2 * xx
      ylmt1 = yfrn1 * yy
      ylmt2 = yfrn2 * yy
      dx = (xlmt2 - xlmt1) / float(iplot - 1)
      dy = (ylmt2 - ylmt1) / float(jplot - 1)

C *** LOCATE SECTION AND DEVELOP INTERPOLATION PARAMETERS
      do 20 i=1,iplot
         do 20 j=1,jplot
            xtmp = xlmt1 + dx * float(i - 1)
            ytmp = ylmt1 + dy * float(j - 1)
            do 10 ii=2,ixx
               if (xxs(ii) .ge. xtmp .and. xxs(ii-1) .lt. xtmp) then
                   iil = ii
               endif
               if (x1(ii) .ge. xtmp .and. x1(ii-1) .lt. xtmp) then
```

```
                ii2 = ii
             endif
   10        continue
             cx = (xxs(ii1) - xtmp) / (xxs(ii1) - xxs(ii1-1))
             cx_s = (x1(ii2) - xtmp) / (x1(ii2) - x1(ii2-1))
             do 15 jj=2,jyy
                if (yys(jj) .ge. ytmp .and. yys(jj-1) .lt. ytmp) then
                   jj1 = jj
                endif
                if (y1(jj) .ge. ytmp .and. y1(jj-1) .lt. ytmp) then
                   jj2 = jj
                endif
   15        continue
             cy = (yys(jj1) - ytmp) / (yys(jj1) - yys(jj1-1))
             cy_s = (y1(jj2) - ytmp) / (y1(jj2) - y1(jj2-1))

             c11_u = cx_s * cy
             c12_u = cx_s * (1. - cy)
             c21_u = (1. - cx_s) * cy
             c22_u = (1. - cx_s) * (1. - cy)

             c11_v = cx * cy_s
             c12_v = cx * (1. - cy_s)
             c21_v = (1. - cx) * cy_s
             c22_v = (1. - cx) * (1. - cy_s)

C *** INTERPOLATION DONE HERE
             uu1(i,j) = c11_u * uu(ii2-1,jj1-1) + c12_u * uu(ii2-1,jj1)
     &                  + c21_u * uu(ii2,jj1-1) + c22_u * uu(ii2,jj1)
             vv1(i,j) = c11_v * vv(ii1-1,jj2-1) + c12_v * vv(ii1-1,jj2)
     &                  + c21_v * vv(ii1,jj2-1) + c22_v * vv(ii1,jj2)
   20   continue

C *** DETERMINE MAXIMUM INTERPOLATED SPEED
        vmax = 0.
        do 30 i=1,iplot
           do 30 j=1,jplot
              vmag = sqrt(uu1(i,j)**2 + vv1(i,j)**2)
              if (vmag .gt. vmax) vmax = vmag
   30   continue
        print *,'THE MAXIMUM SPEED IS',vmax

C *** DETERMINE MAXIMUM INTERPOLATED VELOCITY COMPONENTS
        umax = 0.
        vmax = 0.
        do 40 i=1,iplot
           do 40 j=1,jplot
              if (abs(uu1(i,j)) .gt. umax) umax = abs(uu1(i,j))
              if (abs(vv1(i,j)) .gt. vmax) vmax = abs(vv1(i,j))
   40   continue
        print *,'MAXIMUM INTERPOLATED U-VEL',umax
        print *,'MAXIMUM INTERPOLATED V-VEL',vmax

        return
        end
*******************************************************************************
```

```
      SUBROUTINE PLOT
*************************************************************************
*                                                                       *
*  This subroutine plots the velocity vectors using the VELVCT routine *
*  of the NCAR Graphics package.                                        *
*                                                                       *
*************************************************************************

      common/data/t(25,25,15),u(25,25,15),v(25,25,15),w(25,25,15),
     &            uu(24,24),vv(24,24),uu1(24,24),vv1(24,24)
      common/lmt2/xx,yy
      common/ugrd/iplot,jplot
      common/sct1/iscn,jscn,kscn,nview
      common/sct2/xscn,yscn,zscn
      common/bl2/x,y,h,tflr,twal,ta,xtime
      common/fctn/xfrn1,xfrn2,yfrn1,yfrn2

      external TAG

      real*8 t,u,v,w,uu,vv,uu1,vv1,xx,yy
      real*8 xscn,yscn,zscn,x,y,h,tflr,twal,ta,xtime
      real*8 xfrn1,xfrn2,yfrn1,yfrn2
      character*5 elev,sec
      character*45 title
      character*4 xl1,xl2,yl1,yl2

      dimension varx(24,24),vary(24,24),spv(2)

C *** SET PARAMETERS REQUIRED FOR NCAR GRAPHICS ROUTINES
      data spv/2*0./

***** CONVERT TO SINGLE PRECISION (NCAR WON'T WORK IN DOUBLE PRECISION) *****

C *** FIRE TIME WHEN DATA WAS TAKEN
      time=xtime*h/1.0

C *** SECTION TO PLOT
      if(iscn.gt.0) then
         scn=xscn*h
         xzoom1=xfrn1*y
         xzoom2=xfrn2*y
         yzoom1=yfrn1*h
         yzoom2=yfrn2*h
      elseif(jscn.gt.0) then
         scn=yscn*h
         xzoom1=xfrn1*x
         xzoom2=xfrn2*x
         yzoom1=yfrn1*h
         yzoom2=yfrn2*h
      else
         scn=zscn*h
         xzoom1=xfrn1*x
         xzoom2=xfrn2*x
         yzoom1=yfrn1*y
         yzoom2=yfrn2*y
      endif
```

```
C *** CONVERT REAL NUMBERS TO CHARACTERS FOR USE IN TITLE
      call TAG(scn,elev)
      call TAG(time,sec)
      call TAG(xzoom1,x11)
      call TAG(xzoom2,x12)
      call TAG(yzoom1,y11)
      call TAG(yzoom2,y12)

C *** DEFINE DIMENSIONS OF PLOT AREA
      xlft = 0.125
      xrgt = xlft + 0.75*xx/2.
      ybot = 0.20
      ytop = ybot + 0.75*yy/2.
      x1 = 24.
      if (kscn.gt.0) then
         y1 = 24.
      else
         y1 = 14.
      endif

C *** VELOCITY COMPONENTS
      do 20 i=1,iplot
         do 20 j=1,jplot
            varx(i,j) = uu1(i,j)
            vary(i,j) = vv1(i,j)
  20  continue

C *** START WITH NCAR GRAPHICS
      call GOPKS (6,0)
      call GOPWK (1,2,1)
      call GACWK (1)

C *** TURN OFF CLIPPING SO CHARACTERS ARE PRINTED OUTSIDE PLOT
      call GSCLIP(0)


C *** DEFINE BOUNDARIES AND SET PERIMETER FOR VECTOR PLOT
      call SET (xlft,xrgt,ybot,ytop,1.,x1,1.,y1,1)
      call PERIM (1,0,1,0)

C *** PLOT VELOCITY VECTORS
      call VELVCT (varx,iplot,vary,iplot,iplot,jplot,0.,0.,-1,0,0,spv)

***** LABEL AXIS AND TITLE PLOT *****
c *** X SECTION
      if(iscn.gt.0) then
         title='Y-Z ELEVATION (X = '//elev//' FT.) AT '//sec//' SEC.'
         call PLCHHQ((xlft+xrgt)/2.+.75,ybot-2.5,title,.015,0.,-1.)
         call PLCHHQ(10.,.5,'BREADTH (Y-DIR)',.01,0.,-1.)
         call PLCHHQ(1.,.5,x11,.01,0.,-1.)
         call PLCHHQ(23.,.5,x12,.01,0.,-1.)
         call PLCHHQ(.5,7.0,'HEIGHT (Z-DIR)',.01,90.,0.)
         call PLCHHQ(.5,1.5,y11,.01,90.,0.)
         call PLCHHQ(.5,13.5,y12,.01,90.,0.)
```

```
C *** Y SECTION
      elseif(jscn.gt.0) then
         title='X-Z ELEVATION (Y = '//elev//' FT.) AT '//sec//' SEC.'
         call PLCHHQ((xlft+xrgt)/2.+.75,ybot-2.5,title,.015,0.,-1.)
         call PLCHHQ(10.,.5,'DEPTH (X-DIR)',.01,0.,-1.)
         call PLCHHQ(1.,.5,x11,.01,0.,-1.)
         call PLCHHQ(23.,.5,x12,.01,0.,-1.)
         call PLCHHQ(.5,7.0,'HEIGHT (Z-DIR)',.01,90.,0.)
         call PLCHHQ(.5,1.5,y11,.01,90.,0.)
         call PLCHHQ(.5,13.5,y12,.01,90.,0.)

C *** Z SECTION
      elseif(kscn.gt.0) then
         title='PLAN VIEW (Z = '//elev//' FT.) AT '//sec//' SEC.'
         call PLCHHQ((xlft+xrgt)/2.+1.5,ybot-2.5,title,.015,0.,-1.)
         call PLCHHQ(10.,.5,'DEPTH (X-DIR)',.01,0.,-1.)
         call PLCHHQ(1.,.5,x11,.01,0.,-1.)
         call PLCHHQ(23.,.5,x12,.01,0.,-1.)
         call PLCHHQ(.5,9.0,'BREADTH (Y-DIR)',.01,90.,-1.)
         call PLCHHQ(.5,1.5,y11,.01,90.,0.)
         call PLCHHQ(.5,23.5,y12,.01,90.,0.)
      endif

C *** FINISHED WITH NCAR GRAPHICS
      call GCLRWK (1,1)
      call GDAWK (1)
      call GCLWK (1)
      call GCLKS

      return
      end
*******************************************************************************
      SUBROUTINE TAG(scn,elev)
*******************************************************************************
*                                                                             *
*  This subroutine converts scn to a character value for use in the           *
*  title of the plot.                                                         *
*                                                                             *
*******************************************************************************

      character*1 e(0:9)
      character*5 elev

      data e/'0','1','2','3','4',
     &       '5','6','7','8','9'/

      if(scn.ge.100..and.scn.lt.1000.) then
         i1=int(scn/100.)
         i2=int((scn-real(i1)*100.)/10.)
         i3=int(scn-(real(i1)*100.+real(i2)*10))
         i4=nint((scn*10.-int(scn*10.))*10.)
         elev=e(i1)//e(i2)//e(i3)//'.'//e(i4)
      elseif(scn.ge.10..and.scn.lt.100.) then
         i1=int(scn/10.)
         i2=int(scn-real(i1)*10.)
```

```
      i3=int((scn-int(scn))*10.)
      i4=nint((scn*10.-int(scn*10.))*10.)
      elev=e(i1)//e(i2)//'.'//e(i3)//e(i4)
   elseif(scn.lt.10) then
      i1=int(scn)
      i2=int((scn-int(scn))*10.)
      i3=int((scn*10.-int(scn*10.))*10.)
      i4=nint((scn*100.-int(scn*100.))*10.)
      elev=e(i1)//'.'//e(i2)//e(i3)//e(i4)
   endif

   return
   end
```

LIST OF REFERENCES

1. Quintiere, J., "A Perspective on Compartment Fire Growth", *Combustion Science and Technology*, vol. 39, pp. 11-54, 1984.

2. *Magazine Propellant Fire Tests*, NAWC TP 7128 (Draft), 1992.

3. Aziz, K., and J.D. Hellums, "Numerical Solution of the Three Dimensional Equation of Motion for Laminar Natural Convection", *The Physics of Fluids*, vol. 10, pp. 314-324, 1967.

4. Mallinson, G.D. and G. De Vahl Davis, "Three Dimensional Numerical Analysis of Transient Natural Convection in a Box, a Numerical Study", *International Journal of Heat and Mass Transfer*, vol. 83, pp.1-31, 1977.

5. Morrison, G.L. and V.G. Tran, "Laminar Flow Structure in Vertical Free Convection Cavities", *International Journal of Heat and Mass Transfer*, vol 21, no. 2, pp. 203-213, 1978.

6. Chan, A.M.C. and S. Banerjee, "Three Dimensional Numerical Analysis of Transient Natural Convection in Rectangular Enclosure", *Journal of Heat Transfer*, vol. 101, no. 1, pp. 1427-1438, 1979.

7. Ozeo, H., K. Fujii, N. Lior and S.W. Churchill, "Long Rolls Generated by Natural Convection in an Inclined, Rectangular Enclosure", *International Journal of Heat and Mass Transfer*, vol. 26, no. 10, pp. 1427-1438, 1983.

8. Rehm, R.G. and H.R. Baum, "The Equations of Motion for Thermally Driven, Buoyant Flows", *Journal of Research of the National Bureau of Standards*, vol. 83, no. 3, pp. 297-308, 1978.

9. Rehm, R.G. and H.R. Baum, "Computation of Fire Induced Flow and Smoke Coagulation", Nineteenth Symposium, Int. of Combustion, Combustion Institute, Pittsburg, PA, pp. 921-931, 1982.

10. Rehm, R.G. and H.R. Baum, "Natural Computation of Large Scale Fire Induced Flows", paper presented at the Eighth International Conference on Numerical Methods in Fluid Dynamics, Aachen, West Germany, 28 June- 2 July 1982.

11. Rehm, R.G. and H.R. Baum, "Calculations of Three Dimensional Buoyant Plumes in Enclosures", *Combustion Science and Technology*, vol. 40, pp. 55-77, 1984.

12. Yang, K.T., J.R. Lloyd, A.M. Kanury, and K. Satoh, "Modeling of Turbulent Buoyant Flows in Aircraft Cabins", *Combustion Science and Technology*, vol. 39, pp.107-118, 1984.

13. Kou, H.S., K.T. Yang and J.R. Lloyd, "Turbulent Buoyant Flow and Pressure Variations Around an Aircraft Fuselage in a Cross Wind Near the Ground", *Fire Safety Science--Proceedings of the First International Symposium*, pp.173-184, 1986.

14. Nicolette, V.F., K.T. Yang and J.R. Lloyd, "Transient Cooling by Natural Convection in a Two-Dimensional Square Enclosure", *International Journal of Heat Transfer*, vol. 28, no. 9, pp. 1721-1732, 1985.

15. Yang, H.Q., K.T. Yang and J.R. Lloyd, "Flow Transition in Laminar Flow in a Three Dimensional Tilted Rectangular Enclosure", *Heat Transfer, Proceedings of the Eighth International Heat Transfer Conference*, vol. 4, pp. 1495-1500, 1986.

16. Yang, H.Q., K.T. Yang and J.R. Lloyd, "Laminar Natural Convection Flow Transition in Tilted Three Dimensional Longitudinal Rectangular Enclosures", *International Journal of Heat and Mass Transfer*, vol. 30, no. 8, pp. 1637-1644, 1987.

17. Yang, H.Q., K.T. Yang and J.R. Lloyd, "Three Dimensional Buoyant Bimodal Flow Transition in Tilted Enclosures", *International Journal of Heat and Fluid Flow*, vol. 9, no. 2, pp. 90-97, 1988.

18. Department of Aerospace and Mechanical Engineering, University of Notre Dame, South Bend, IN, Technical Report, TR-37191-74-4, "A Numerical Model for the Prediction of Two Dimensional Unsteady Flows of Multicomponent Gases with Strong Buoyancy Effect and Recirculation", by Michael L. Doria, November 1974.

19. Leonard, S.P., "A Convectively Stable, Third-Order Accurate Finite- Difference Method for Steady Two-Dimensional Flow and Heat Transfer", *Numerical Properties and Methodologies in Heat Transfer*, ed. T.M. Shih, Hemisphere Publishing Corp., Washington, DC, pp. 211-226, 1983.

20. Nies, G.F., *Numerical Field Model Simulation of Full Scale Tests in a Closed Vessel*, Master's and Mechanical Engineer's Thesis, Naval Postgraduate School, Monterey, CA, December 1986.

21. Raycraft, J.K., *Numerical Field Model Simulation of Full Scale Fire Tests in a Closed Spherical Cylindrical Vessel*, Master's and Mechanical Engineer's Thesis, Naval Postgraduate School, Monterey, CA, December 1987.

22. Houck, R.R., *Numerical Field Model Simulation of Full Scale Fire Tests in a Closed Spherical Cylindrical Vessel with Internal Ventilation*, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1988.

23. McCarthy, T.G., *Numerical Field Model Simulation of Full Scale Fire Tests in a Closed Spherical Cylindrical Vessel Using Advanced Computer Graphics Techniques*, Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1991.

24. *NCAR Graphics User's Guide, Version 2.00*, National Center for Atmospheric Research, Scientific Computing Division, Boulder, CO, 1987.

25. *NCAR Graphics Guide to New Utilities, Version 3.00*, National Center for Atmospheric Research, Scientific Computing Division, Boulder, CO, 1989.

26. Patankar, S.V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Company, New York, NY, 1980.

27. Yang, H.Q., *Laminar Buoyant Flow Transitions in Three Dimensional Tilted Rectangular Enclosures*, Ph.D. Thesis, University of Notre Dame, South Bend, IN, 1987.

28. Department of Aerospace and Mechanical Engineering, University of Notre Dame, South Bend, IN, Technical Report TR-79002-78-2,*An Algebraic Turbulence Model for Buoyant Recirculating Flow*, by V.W. Nee and V.K. Liu, 1978.

29. Sparrow, E.M. and R.D. Cess, *Radiation Heat Transfer*, Hemisphere Publishing Corporation, Washington, DC, 1978.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code ME Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Naval Engineering Cirricular Office Code 34 Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Professor M.D. Kelleher, Code ME kk Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943-5000	2
6.	Dr. Patricia Tatum Code 6183 Naval Research Laboratory Washington, DC 20375	1
7.	Mr. James Gagorik Code 211, ONT Balston Towers #1 800 N. Quincy Arlington, VA 22217	1
8.	Commandant (G-MTH-2) United States Coast Guard 2100 2nd St. SW Washington, DC 20593	1
9.	Coast Guard Law Library United States Coast Guard 2100 2nd St. SW Washington, DC 20593	1

10. Professor K.T. Yang
Department of Aerospace and
Mechanical Engineering
University of Notre Dame
Notre Dame, IN 46556 — 1

11. Mr. Kent Farmer
Code 38903, Bldg 31618
Naval Air Warfare Center
China Lake, CA 93555 — 1

12. LT Kenneth J. Thorkildsen
U.S. Coast Guard Marine Safety Center
400 7th Street, SW
Washington, DC 20590-0001 — 1

846-217

Thesis
T4566 Thorkildsen
c.1 Numerical field model
simulation of fire and
heat transfer in a
rectangular compartment.

AMERICAN BINDERY 2 9-93

3 2768 00035924 4